GNSS Tolling Solution Blueprint

Solution Design
EETS Integration Guide
Doc No. 30006
Version 0.1

Author:
Roman Trinko

X
Max Mustermann

Release:
Hans Gidoff

X
Mimi Musterfrau

Overview of changes.

No.	Version	Status	Date	Contributor	Type of the change
1	0.1	Draft	17.4.2026	Paul Litzinger	First Version
2					
3					

Table 1 Overview of changes

Reference to the status, versions and data classification.

Status:	
Draft	the document is being processed
Released	the document has been checked and released, it can only be modified if the version number is updated.
Obsolete	the document is not valid anymore
Versions:	
0.1, 0.2	draft versions
1.0	**first released version with the status "Released"**
1.1, 1.2, etc.	draft versions, that supplement version 1.0
2.0	**second released version with the status "Released"**
Data classification	
Public	No restriction
Internal	Restricted to internal and external Kapsch employees
Confidential	**Restricted to selected active directory and/or sharepoint groups (default)**
Secret	Restricted to selected employees, server encryption needed

Page

Table of Contents

List of Figures

List of Tables

1 Document purpose

This document describes requirements and considerations for EETS Integration, as well as the data objects exchanged between the toll charger and the TSP based on the EFC (Electronic Fee Collection) toolbox standard ISO 12855. The EETS TSPs integrate with the solution based on "Electronic fee collection — Information exchange between service provision and toll charging" ISO 12855:2025(en) standard and corresponding "Electronic fee collection — System architecture for vehicle-related tolling Part 3: Data dictionary" ISO/TS 17573-3:2021 standard. The base ASN.1 specifications are available under ISO 12855 ASN.1 and ISO 17575 ASN.1. Since there is no corresponding released profile ISO 16986 version available at the time of writing this document, the solution aligns on the available draft version. Right to change is reserved.
The document provides a general description of the solution and its configurations. Project specific configurations overrule general aspects as defined in the Configurations and Definitions section of this document.

2 Data exchange

The data objects described in this chapter are exchanged between the following two entities:

- Toll Charger (TC), and
- Toll Service Provider (TSP) - also referred to as EETS TSP

The solution distinguishes between two message flows:

- incoming: a request is sent from EETS TSP to the TC
- outgoing: a request is sent form the TC to any EETS TSP

In line with the ISO standard, the data are exchanged based on Info Exchange Messages which contains APCI and ADU Information.

2.1 Message Exchange

The messages are exchanged using via a HTTP WebService which exposes and endpoint to which sendAPDU messages are transmitted using the POST method. In line with the ISO 12855 standard, the root object for message exchange is the "infoExchange" object. This object, which is also called Application Protocol Data Unit, contains:

- a single occurrence of Application Protocol Control Information (APCI) object, including information about the sender, receiver, identifier, etc
- one or more Application Data Unit (ADU) objects of the same type.

The following ADUs are supported for automatic exchange via the EETS interface. Details on data objects, formats and value range restrictions are available in the interface specification.

Application Data Object	Direction	Originator	Exchange Mechanism
Exception Lists ADU	incoming	EETS TSP	HTTP WebService
Toll Declarations ADU	incoming	EETS TSP	
Billing Details ADU	outgoing	TC	
Payment Claims ADU	outgoing	TC	

Application Data Object	Direction	Originator	Exchange Mechanism
Acknowledgement ADUs	incoming	EETS TSP	
	outgoing	TC	

Table 2 Data Exchanged

To establish trust between the TC and EETS TSP solution, a trust object ADU needs to be exchanged. This exchange is done manually. No other data object is exchanged. The following diagram illustrates the high level message exchange process, including the sending of a message and the asynchronous acknowledgement response. The flow depicted shows an example where the TC receives a message and answers with an acknowledgement ADU, a typical example is the Toll Declaration ADU.

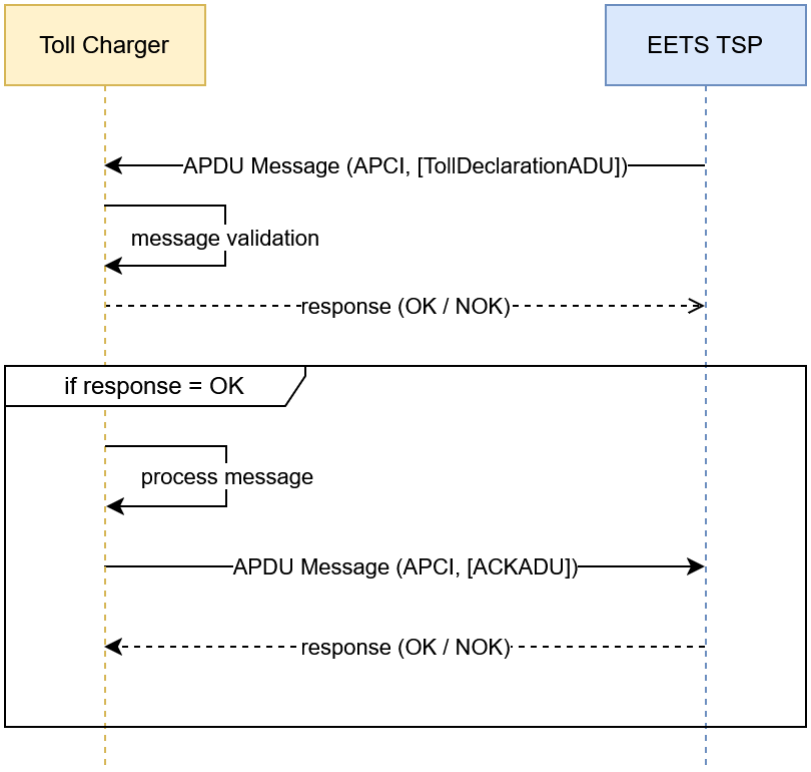

Figure 1 EETS High Level Message Exchange

As depicted, only messages passing message validation process successfully will be further processed by the solution. Therefore, successful transmission of the data is not sufficient for the originator to conclude that the data-sets and are correct according to all business rules and have been accepted by the recipient. The technical acceptance process which is executed asynchronously may detect the violation of a business rule and result in the rejection of the data. Depending on the result of the process, the asynchronous acknowledgement (ACK) ADU will signal business acceptance or rejection of the message. The achnowledgement ADU transmitted in this process contains specific information in case technical errors or warnings have occurred during the processing for the data.

2.1.1 General message considerations

- The solution uses JSON Encoding Rules (JER)
- All reported times are interpreted in generalized time and in UTC +/- 0. The required format is "YYYYMMDDHHmmssZ".
- In case of a transmission error (e.g.: TSP system is unavailable) the systems retry to send the message. In case the message cannot be delivered for an extended period of time, the receiving entity shall be informed and further steps shall be mutually agreed.

The following chapters describe the different Application Data Unit types in detail.

2.2 Toll declaration

This section describes the process and requirements for Toll Declaration exchange.

2.2.1 Business Context

The collection process uses a GNSS device to record the location information. A GNSS device is a single device, capable of recording GNSS location information and transmitting them to the central tolling system. The GNSS device is mounted inside a toll liable vehicle to record the vehicles movement. The GNSS device records the locations (i.e., GNSS positions) of the vehicle according to defined and configurable frequency - *position storing interval* - (e.g., every 5 seconds) when inside the charging relevant area - *toll domain area* . For each position, the GNSS device records the following information to fulfill solutions needs:

- longitude,
- latitude,
- altitude (optional),
- date and time, and
- the following quality parameters:
  - speed
  - course
  - horizontal accuracy

Optionally, the devices may implement a stand still filter, to prevent the collection of position clouds in congestion situations or while the vehicle is waiting at a red light. The collected location information is enriched with charging relevant vehicle attributes and aggregated according to the defined and configurable aggregation period - *position reporting frequency* - (e.g., 5 minutes). The *charging relevant vehicle attributes* are a toll domain specific set and may vary between toll domains based on the used tariff table. Amongst others, the solution supports the following typical attributes:

- Vehicle class: according to Vehicle registration documents
- Vehicle axles: tractor and trailer axles according to vehicle registration documents and current vehicle train configuration
- Vehicle weight limits: according to vehicle registration documents
- Euro class: according to vehicle registration document
- CO2 emission class: according to the EU scheme of CO2 emission classes based on vehicle registration documents

To identify the vehicle and the device used for collection, the packaged position data are attributed with following identifiers:

- License plate number: the country code and LPN according to vehicle registration documents
- Device identifier: the unique identifier of the the GNSS device used to record the location information, composed of provider, manufacturer and equipment identifier

The resulting position package is transmitted to the central tolling system for the detection of the use of toll liable roads and the calculation of the related toll charges.

2.2.2 Communication Process

The charging process starts with the exchange of position packages - *Toll Declaration ADUs*, compiled as previously described. The solution receives Toll Declaration ADUs (TD ADUs), packaged in Info Exchange messages in the agreed aggregation level (*maximum number of TD ADUs)*, from accredited EETS TSP. After successful syntactical and semantic validation of the TD ADUs, the solution response with an asynchronous Acknowledgement ADU (ACK ADU), confirming the reception and successful processing of all provided TD ADUs. In case errors are encountered during validation, the solution rejects individual TD ADUs with a meaningful error code. The validations performed include but are not limited to:

- Syntactical correctness (Error code: apduNotOk)
- Syntactical correctness on TD ADU (Error code: apduNotOk)
- Duplicate transmission of TS ADUs (Error code: tollDeclarationDuplicateTollDeclarationRejected)

Internal errors can be caused by too large messages (e.g.: list of TD ADUs in a request was outside agreed value range). Only successfully acknowledge TD ADUs will be subject to charge evaluation. The exchange follows the communication and timings as defined in ISO 16986.

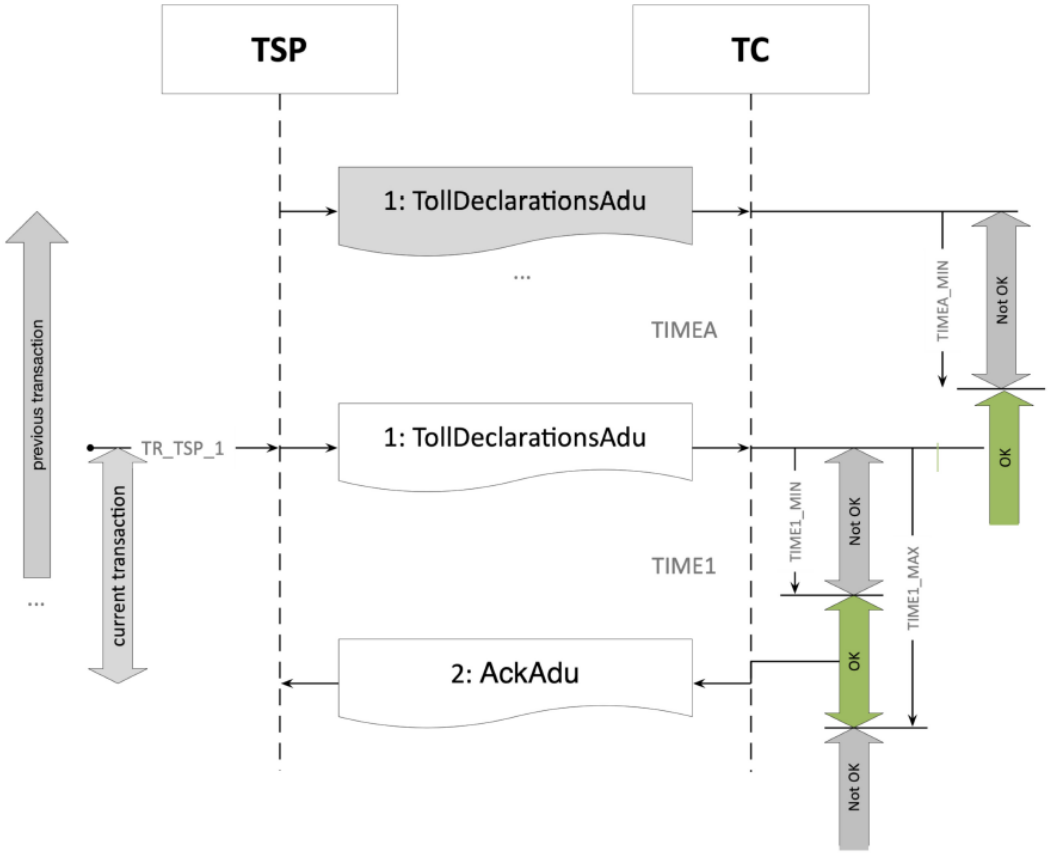

Figure 2 Toll Declaration Exchange

With the following configurable timings:

Trigger/Timing	Meaning	Default
TR_TSP_1	A TSP may initiate a TOLLDECLARATION_SECT transaction at any time after TIMEA_MIN.	Default = 30 minutes, according to the agreed reporting interval per OBE
TIMEA_MIN	The minimum time between the sending of two TollDeclarationAdus in two consecutive TOLLDECLARATION_SECT transactions.	Default = 0 seconds to allow for list of TD ADUs
TIME1_MIN	The minimum time required by the TC to acknowledge the received toll declaration by sending an ACK ADU.	Default = 0 seconds to allow of immediate rejection in syntax error case
TIME1_MAX	The maximum time allowed to the TC to acknowledge the received toll declaration by sending an ACK ADU.	Default = 24 h. To take into account operational procedures in case of maintenance or unplanned downtime. In normal operations the response should be within a few (10) seconds.

Table 3 Toll Declaration Timings

2.2.3 Additional Requirements

- Every transmitted and positively acknowledged TD ADU is subject to charging. The solution does not evaluate the corresponding user exception list entry.
- An APDU containing TD ADUs, which is fully or partially rejected by an ACK ADU must not be re-sent. It is expected that the EETS TSP creates a new APDU containing corrected TD ADUs to allow processing.
- Processing of Toll Declaration data containing future GNSS time stamps is not supported. Such TDs will be acknowledged by the solution and silently discarded by the map matching process.

2.3 Billing details

This section describes the process and requirements for Billing Details exchange.

2.3.1 Business Context

The solution identifies tolled road usage for the received TD ADUs using a map matching process employing a gap closing algorithm. The resulted charges are grouped into single aggregated tariff classes, which are the base for Billing Detail creation. The Billing Details are created by the solution on user level and based on a configurable aggregation interval (*billing detail aggregation period)*. During aggregation, the solution groups all detected charge objects into single tariff class sessions. In case a use is subject to different tariff classes within the billing detail aggregation period, the system will create one billing detail per distinct tariff class. A billing detail contains the following information for each user:

- Period: the begin and end equal to billing detail aggregation period
- Billing detail amount
- Identified tariff
- Vehicle description
- External costs

- Array of used sections with corresponding identifiers and used distance

The Billing Details are expected to be accepted or rejected by the EETS TSP. Only accepted Billing Details will be further processed. In case of Billing Detail rejection, an operations process is followed. During bilateral discussion and after reaching an agreement a new billing detail will be issued with corrected amount and referencing the billing detail subject to initial rejection.

2.3.2 Communication Process

The created Billing Details are sent to the EETS TSP as Billing Details ADU upon completion. The EETS TSP validates the Billing Details and signals acceptance or rejection in ACK ADUs. Only positively accepted Billing Details will be subject to further processing. Rejected Billing Details are reported to the operations team and are clarified with EETS TSP. In case of rejection of a BillingDetail, the TSP informs the reason of the rejection. Irrespective of the reason, the Billing Detail is considered rejected and needs to be clarified with the TSP. The following reasons and related codes are supported by the solution:

- billingDetailsIssuerIdRejected: code 700
- billingDetailsRejected: code 701
- billingDetailsTollContextOperatorMismatch: code 702
- billingDetailsTollContextOperatorRejected: code 703
- billingDetailsUserIdRejected: code 704
- billingDetailsPeriodRejected: code 705
- billingDetailsAmountRejected: code 706
- billingDetailsUserIdLpnMissing: code 719
- billingDetailsUserIdLpnNotResponsible: code 720
- billingDetailsUserIdObeIdMissing: code 724
- billingDetailsPeriodMissing: code 730
- billingDetailsContentDupicate: code 735

The following reason codes are **not** supported by the solution.

- billingDetailsContextNameRejected: code 707
- billingDetailsAppliedUserClassRejected: code 708
- billingDetailsAppliedLocalVehicleClassRejected: code 709
- billingDetailsAppliedTimeClassRejected: code 710
- billingDetailsEntryTimeRejected: code 711
- billingDetailsEntryChargeObjectRejected: code 712
- billingDetailsEntryLocationClassRejected: code 713
- billingDetailsExitChargeObjectRejected: code 714
- billingDetailsExitTimeRejected: code 715
- billingDetailsReferenceRejected: code 716
- billingDetailsExitLocationClassRejected: code 717
- billingDetailsBillingDetailsInfoNotSupported: code 718
- billingDetailsUserIdPanMissing: code 721
- billingDetailsUserIdPanNotValid: code 722

- billingDetailsUserIdPanNotAllowed: code 723
- billingDetailsUserIdEfcContextMarkMissing: code 725
- billingDetailsUserIdEfcContextMarkNotAllowed: code 726
- billingDetailsPaymentMeansIgnored: code 727
- billingDetailsRelatedBillingDetailsUnknown: code 728
- billingDetailsRelatedBillingDetailsInfoNotSupported: code 729
- billingDetailsAmountIgnored: code 731
- billingDetailsRefTollDeclarationIgnored: code 732
- billingDetailsRefTollDeclarationUnknown: code 733
- billingDetailsRefTollDeclarationIdNotSupported: code 734

The exchange follows the triggers and timing as defined in ISO 16986

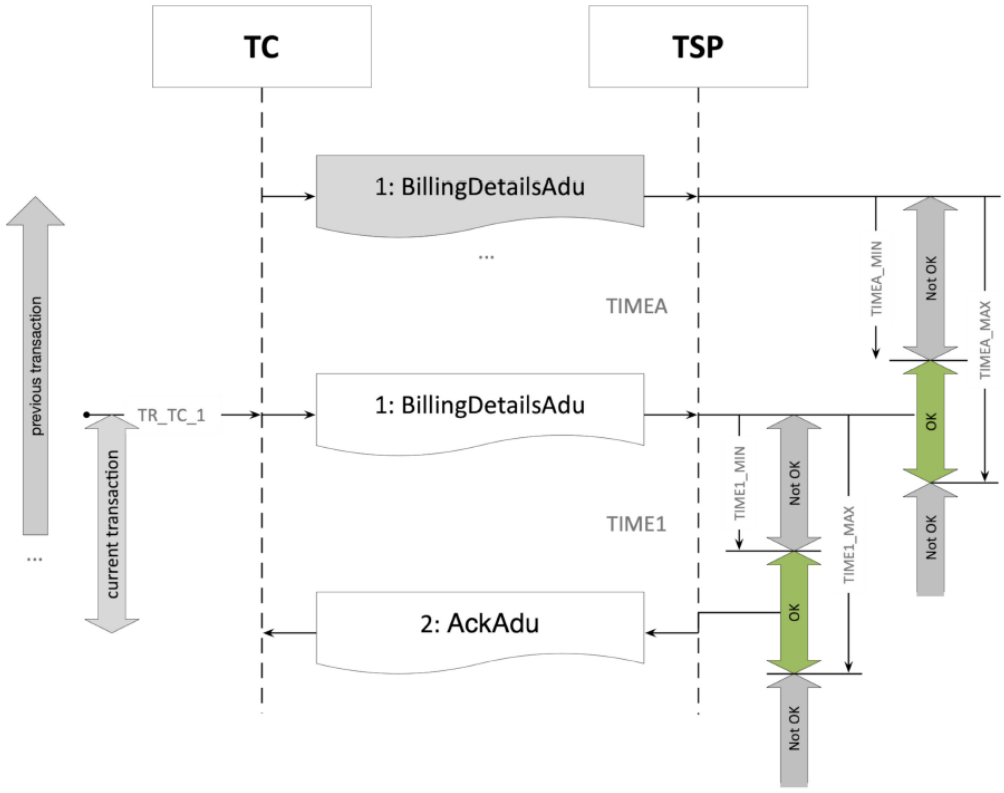

Figure 3 Billing Details Exchange

With the following configurable timings

Trigger/Timing	Meaning	Default
TR_TC_1	A TC shall initiate a BILLINGDETAILS_TC at any time in the interval between TIMEA_MIN and TIMEA_MAX.	according to grouping time, default BD creation starts at 01:00:00 covering the charges of the previous period.
TIMEA	The time between the sending of two BillingDetailsAdus in two consecutive BILLINGDETAILS_TC transactions.	Default = 0 seconds, to allow list of BD ADUs

Trigger/Timing	Meaning	Default
TIMEA_MIN	The time between the sending of two BillingDetailsAdus in two consecutive BILLINGDETAILS_TC transactions.	Default = 0 seconds, to allow list of BD ADUs
TIMEA_MAX	The maximum time for sending of two BillingDetailsAdus in two consecutive BILLINGDETAILS_TC transactions	Default = 24 hours, to allow for interfering of operations due to planned or unplanned interruptions
TIME1_MIN	The minimum time required by the TSP to acknowledge the BillingDetailsAdu by sending an AckAdu.	Default = 0 seconds, to allow of immediate rejection in syntax error case
TIME1_MAX	The maximum time allowed to the TSP to acknowledge the BillingDetailsAdu by sending an AckAdu.	Default = 6 hours, to ensure that the processing of AckAdu can be completed before the next cycle

Table 4 Billing Details Timings

Billing Detail which are not acknowledged within the agreed timings are considered as rejected by the solution. Delayed transmission of ACK ADUs are rejected by the solution.

2.3.3 Additional Requirements

No additional requirements.

2.4 Payment claims

This section describes the process and requirements for Payment Claims exchange.

2.4.1 Business Context

The final stage of the EETS charging process is to issue Payment Claims. Similar to Billing details which group usage statements, th solution groups Billing Details into Payment Claims on a configurable periodicity. Only positively acknowledged Billing Details will be grouped. The grouping is subject to configurable business rules, such as grouping by time, grouping by maximum number of Billing Details, etc.
The following grouping is default:

- *payment claim aggregation period*: default 30 days (1st day of the month at 00:00:00 to last day of the month at 23:59:59) (global configurable value)
- *maximum number of billing details*: default 5000 (global configurable value) to restrict the maximum size of a payment claim

As payment claims are not grouped on user level, they contain the following minimum information:

- total amount
- aggregation start and end
- referencing the acknowledged billing details as ReferencedAduIdentifier.

In general, it is not expected that Payment Claims are rejected, as they include only acknowledged Billing Details. In case of Payment Claim rejection, an operations process is followed. During bilateral discussion and after reaching an agreement, a new Payment Claim will be issued with the corrected amount and referencing the payment claim subject to initial rejection.
All positively acknowledged Payment Claims will be subject to payment by the EETS TSP.

2.4.2 Communication Process

The created Payment Claims are sent to the relevant EETS TSP as Payment Claims ADU in the agreed frequency and aggregation. The EETS TSP validates the Payment Claims and signals acceptance or rejection in ACK ADUs. Only positively accepted Payment Claims will be subject to further processing (invoice generation, etc). Rejected Payment Claims are reported to the operations team and are clarified with EETS TSP. The timings presented are considered configurations and might change according to the needs of the solution, for example due to performance considerations. It is expected that the TSP solution can adapt to the changes required. Payment Claims which are not acknowledged within the agreed timings are considered as rejected by the solution. Delayed transmission of ACK ADUs are rejected by the solution.
In case of rejection of a Payment Claim, the TSP informs the reason of the rejection. Irrespective of the reason, the Payment Claim is considered as rejected and needs to be clarified with the TSP. The following reasons are supported by the solution:

- claimRejectedByTsp: code 800
- paymentClaimIdRejected: code 802
- paymentClaimStartDateTimeRejected: code 803
- paymentClaimEndDateTimeRejected: code 804
- paymentClaimUserIdRejected: code 805ö
- paymentClaimPaymentClaimAmountRejected: code 806
- paymentClaimPaymentClaimStatusRejected: code 807
- paymentClaimTypeOfContentRejected: code 808
- paymentClaimEndDateTimeMissing: code 809
- paymentClaimStartAndEndDateTimeInvalid: code 810
- paymentClaimUserIdMismatch: code 812
- paymentClaimUserIdUnknown: code 813
- paymentClaimUserIdBlocked: code 814
- paymentClaimUserIdNotSupported: code 815
- paymentClaimUserIdMissingForCreditNote: code 816
- paymentClaimPaymentClaimAmountMismatch: code 817
- paymentClaimPaymentClaimStatusNotSupported: code 818
- paymentClaimTypeOfContentMissing: code 819
- paymentClaimTypeOfContentNotSupported: code 820
- paymentClaimReferenceDetailListMismatch: code 822
- paymentClaimDetailsPaymentDetailTypeMissing: code 828
- paymentClaimDetailsPaymentDetailTypeNotSupported: code 829
- paymentClaimDetailsPaymentDetailTextMissing: code 830
- paymentClaimDetailsPaymentDetailQuantityMissing: code 831
- paymentClaimDetailsPaymentDetailQuantityNotSupported: code 832
- paymentClaimDetailsPaymentDetailDiscountMissing: code 833
- paymentClaimDetailsPaymentDetailBasicAmountMissing: code 834
- paymentClaimDetailsPaymentDetailBasicAmountMismatch: code 835

- paymentClaimDetailsPaymentDetailAmountExclVatMissing: code 836
- paymentClaimDetailsPaymentDetailAmountExclVatMismatch: code 837
- paymentClaimDetailsPaymentDetailAmountTotalMissing: code 838
- paymentClaimDetailsPaymentDetailAmountTotalMismatch: code 839

The following reason codes are **not** supported by the solution.

- paymentClaimUserIdMissingForClaim: code 811
- paymentClaimReferenceDetailListMissing: code 821
- paymentClaimReferenceDetailListIgnored: code 823
- paymentClaimPaymentReferenceIgnored: code 824
- paymentClaimPaymentClaimDetailsIgnored: code 825
- paymentClaimDetailsPaymentDetailCodeIgnored: code 826
- paymentClaimDetailsPaymentDetailNumberIgnored: code 827

The exchange follows the communication and timings as defined in ISO 16986

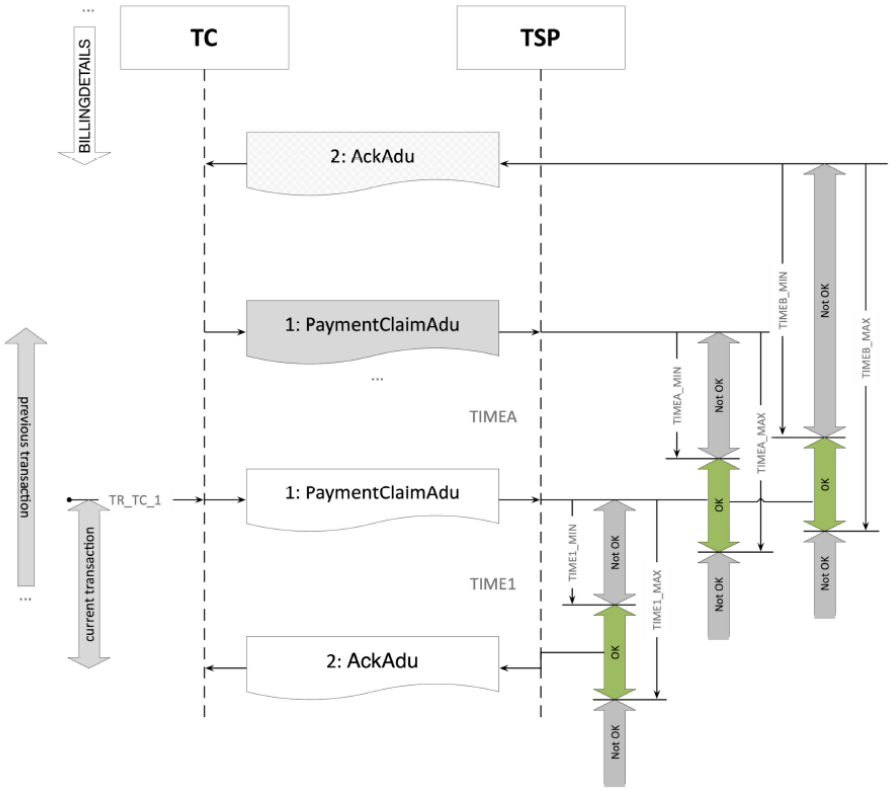

Figure 4 Payment Claims Exchange

With the following timings:

Trigger/Timing	Meaning	Default
TR_TC_1	A PAYMENTCLAIM transaction shall be initiated within the intersection of the time interval between TIMEA_MIN and TIMEA_MAX with the time interval between TIMEB_MIN and TIMEB_MAX.	The payment Claim will be sent within a configurable time period of the grouping interval according to grouping time, default PC creation starts at 01:00:00 covering the charges of the previous period
TIMEA_MIN	The minimum time between the sending of two PaymentClaimAdus in two consecutive PAYMENTCLAIM transactions	Default = 0 seconds, to enable sending a list of Payment Claims
TIMEA_MAX	The maximum time between the sending of two PaymentClaimAdus in two consecutive PAYMENTCLAIM transactions	Default = 20 days, to account for outages and clarifications of not acknowledged Billing details
TIMEB_MIN	The minimum time between the sending of a PaymentClaimAdu in a PAYMENTCLAIM transaction and AckAdu in the corresponding BILLINGDETAILS transaction.	Default = 0 min, to enable same day issuing of Payment Claims
TIMEB_MAX	The maximum time between the sending of a PaymentClaimAdu in a PAYMENTCLAIM transaction and AckAdu in the BILLINGDETAILS transaction transferring a BillingDetailsAdu referenced in the PaymentClaimAdu.	Default = 20 days, to account for outages and clarifications of not acknowledged Billing details
TIME1_MIN	The minimum time required to acknowledge the received PaymentClaimAdu in a PAYMENTCLAIM transaction.	Default = 0 seconds, to allow of immediate rejection in syntax error case
TIME1_MAX	The maximum time required to acknowledge the received PaymentClaimAdu in a PAYMENTCLAIM transaction.	Default = 6 hours, to ensure that the processing of AckAdu can be completed before the next cycle

Table 5 Payment Claims Timings

2.4.3 Additional Requirements

No additional requirements

2.5 Exception lists

This section describes the process and requirements for Billing Details exchange.

2.5.1 Business Context

EETS TSP lists are used to support the enforcement process or to identify EETS TSPs which shall be notified in case a business process demands a notification. TSP lists are exchanged in accordance with the ISO12855:2025 message structure. Due to the expected list size of TSP Lists, full lists are only exchanged during night times. The system supports the following list types:

- Access list: A list containing all users the TSP has a contract with and for which the TSP guarantees toll payment. The LPN is a mandatory data field in this list. The access list needs to be accessible for business process.
- Block list: A list containing all users the TSP has a contract with, but for which the guarantee for toll payment has temporarily been suspended. The LPN is a mandatory data field in this list. A user can only be removed from a block list explicitly (not via adding to the access list).

Block and access list are not mutually exclusive. The system supports only full list exchange. To limit the size of the lists, only the following minimal information is exchanged:

- list type
- list version: needs to be incremented, is unique in combination with list type
- list validity start
- for each list entry:
  - License Plate Number
  - Obe identifier
  - Payment Means (mandatory due to profile)
  - reason: for access lists only reason 8 (user on access list) is supported.

A user on a list is considered to be on the list until the user is removed from the list at the next possible list exchange.

2.5.2 Communication Process

The list exchange process starts with the exchange Exception List (EL) ADUs compiled by the EETS TSP. The system receives Exception List ADUs packaged in Info Exchange messages transmitted at agreed time window (*full exception list exchange window)* from accredited EETS TSP. Due to the expected EL size, the system only allows the transmission of one EL ADU within an Info Exchange message. Additionally, EL ADUs are discriminated by exception list type. Only one EL type can be exchanged at a time (for example Access Full List or Block Full List). After successful syntactical and semantic validation of the EL ADUs, the system response with an asynchronous Acknowledgement ADU confirming the reception and processing of the EL ADU. In case of errors encountered during validation, the system rejects the EL ADU with a meaningful error code.
The validations performed include but are not limited to:

- syntax error (apduNotOk)
- syntax error on EL ADU (apduNotOk)
- exception list type rejected, rejection reason code 401

Only successfully acknowledge EL ADUs will be subject to charge evaluation. The exchange follows the communication and timings as defined in ISO 16986.

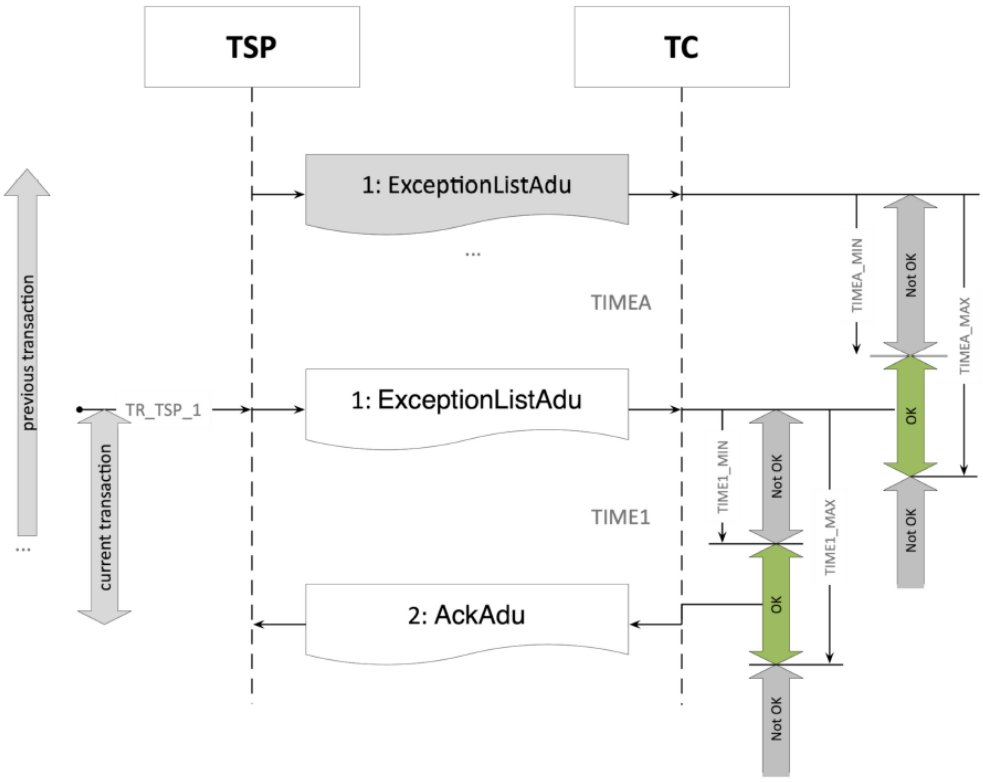

Figure 5 Exception List Exchange

With the following timings

Trigger/Timing	Meaning	Default
TR_TSP_1	This trigger shall be initiated within the time interval defined by TIMEA_MIN and TIMEA_MAX.	
TIMEA_MIN	The minimum time between sending of two ExceptionListAdus in two consecutive EXCEPTIONSLIST transactions	Default = 10 minutes
TIMEA_MAX	The maximum time allowed between sending an ExceptionListAdu in two consecutive EXCEPTIONLIST transactions.	Default = 300 minutes, constraint by the *full exception list exchange window*
TIME1_MIN	The minimum time required to acknowledge the provided ExceptionListAdu by sending an AckAdu.	Default = 0 seconds to allow of immediate rejection in syntax error case
TIME1_MAX	The maximum time allowed to acknowledge the provided ExceptionListAdu by sending an AckAdu.	Default = 24 h. To take into account operational procedures in case of maintenance or unplanned downtime. In normal operations the response should be within a few (10) seconds.

Table 6 Exception List Timings

2.5.3 Additional Requirements

- In case a user is on the block list, the EETS TSP is expected to not send Toll Declaration ADUs for this user as long as the user is on the block list.
- In case a user is on the access list, the EETS TSP is expected to send Toll Declaration ADUs, unless the user is included in the block list as well.
- In case the validity start of the list is in the past, the system considers it to be active with the data and time of reception.
- In case a not expected reason is provided as reason code, the system assumes that the reason code is erroneous but accepts the list entry.
- In case a EL ADU could not be successfully exchanged during the *full exception list exchange window* the iteration of EL exchange is skipped. Note: This is possible as the successful exchange of the list is not influencing the charging behavior.
- The EETS TSP is responsible to keep the lists up to date in accordance with the allowed update interval.

2.6 Acknowledgements

This section describes the process and requirements for Acknowledgement exchange.

2.6.1 Business Context

The solution exchanges Acknowledgements (ACK) ADUs in order to complete a business process. The ACK ADUs contain the business information about the acceptance or rejection of the referenced ADU. The ACK ADU is can be used to signal complete acceptance or rejection of an entire APDU or to inform only individual ADU(s) rejection, providing additional reasons. In general the following logic holds:

Field	Value and Usage
apduAckCode	• If the apduAckCode is apduOk, and the issues structure is omitted, or transported empty, then all ADUs are accepted. • If the apduAckCode is apduOK, but the issues structure is present, and it contains issues, then the issues listed with an issueCode are rejected. • If the apduAckCode is not apduOK then the whole message is rejected, including all of its ADUs.
issues	this object identifies the specific ADU identifiers in case of partial rejection.
issueCode	Must be set to one of the values as supported in the corresponding business context

2.6.2 Communication Process

The communication of Ack ADUs can be triggered by both parties, TC and EETS TSP. Whoever creates the ACK ADUs sends it immediately to the receiving entity. In line with ISO 16986 profile no timings or trigger restrictions are specified.

2.6.3 Additional Requirements

No additional requirements

3 General Requirements, Obligations and Considerations

This section describes general requirements and obligations that EETS TSPs are expected to fulfill as well as considerations EETS TSPs need to take into account.

3.1 Security Aspects

The following table gives an overview of the layered scurity setup:

Layer	What It Protects	When It Protects	Why It Matters
TLS	Transport confidentiality & integrity	During every request	Prevents eavesdropping and tampering in transit
OAuth 2.0 +ClientId + Client Secret	Client authentication	When requesting tokens	Ensures only trusted clients can obtain JWTs
JWT Access Token	Authorization & identity	On every API call	Enforces permissions and validates caller identity
Refresh Token	Session continuity	When access token expires	Enables long sessions without exposing API key
Payload Signing	Message integrity	On every signed request	Ensures payload cannot be altered without detection

Table 7 Layered Security Architecture

When a client interacts with the system, security is enforced in layered stages. Each mechanism protects a different part of the communication flow. TLS provides encrypted transport so data cannot be intercepted or modified in transit. The identity verification step happens when the client presents its client Id and secret to the OAuth 2.0 token endpoint, prooving it is an authorized application and receiving a short-lived JWT in return. The JWT then becomes the client's access token for executing business needs. JWT being validated on every request to ensure the caller is authenticated and authorized. JWTs expire quickly to limit exposure, refresh tokens allow the client to obtain new access tokens without repeatedly presenting the API key. Finally, payload signing adds message-level integrity: the client signs the request body with its private key, and the system verifies the signature using the corresponding public key, ensuring the payload itself has not been altered. Together, these layers form a defense-in-depth model that secures the transport, authenticates the client, authorizes each request, maintains long-lived sessions safely, and guarantees that the payload remains intact.

- **TLS** Protects the *transport channel* by encrypting data in transit and ensuring the client is talking to the legitimate server. It prevents eavesdropping and tampering during transmission but does not authenticate the client or authorize API access.
- **OAuth 2.0 + clientId and secret** Protects the *token issuance stage*. The API key identifies the client application and allows it to obtain a JWT. This ensures only registered and trusted clients can request tokens.
- **JWT Access Token** Protects *each API request*. The JWT carries identity and authorization claims and is validated by your API gateway or backend on every call. It ensures the caller is authenticated and allowed to perform the requested action.

- **Refresh Token** Protects *session continuity*. It allows the client to obtain new access tokens without repeatedly exposing the API key. This reduces risk while enabling long-lived sessions.
- **Payload Signing** Protects *message integrity*. The client signs the payload with its private key, and the server verifies the signature using the client's public key. This ensures the payload has not been altered, even if the transport layer is secure.

The following diagram gives an overview of the security setup.

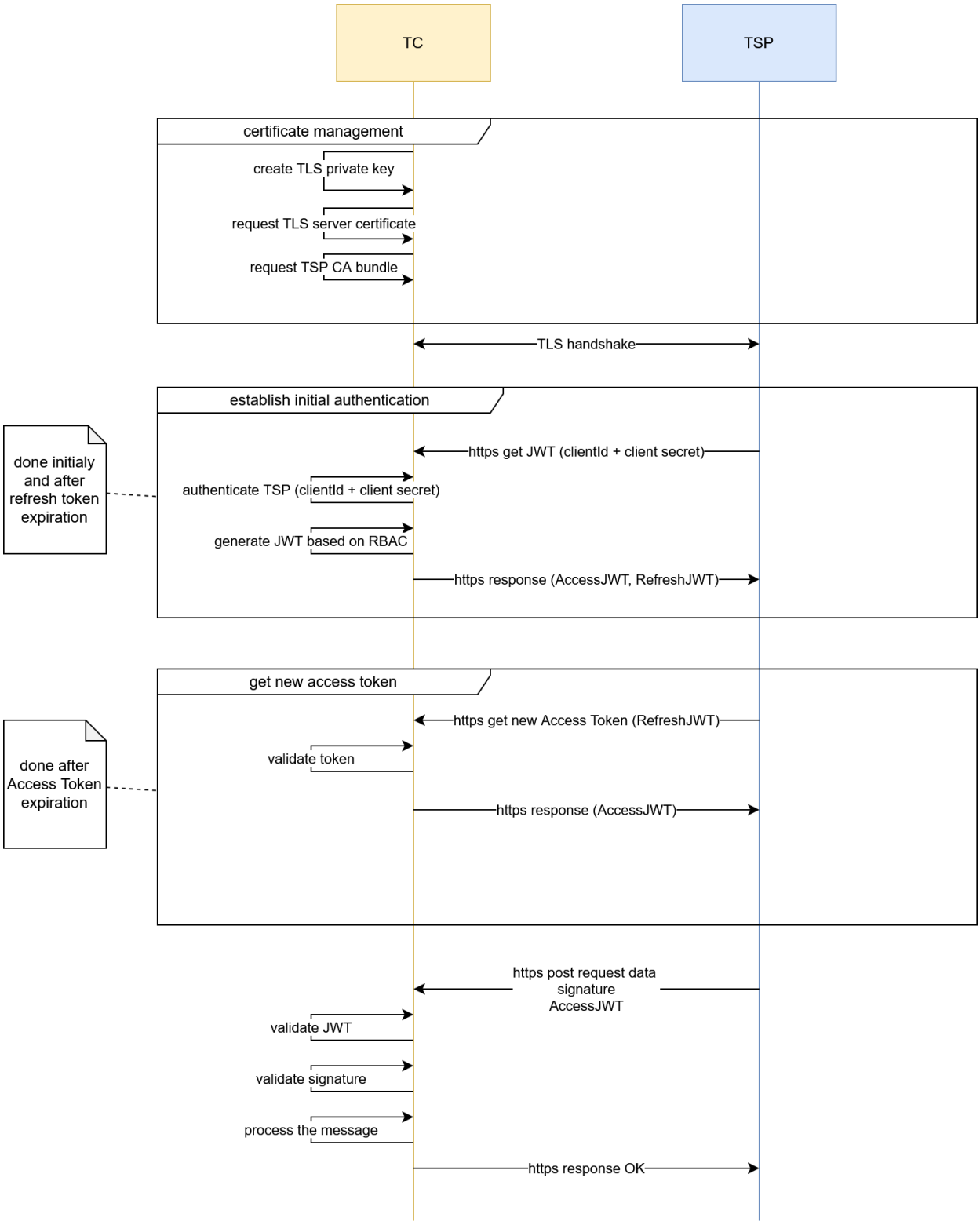

Figure 6 Security Communication Flow

The selected security architecture represents a state of the art architecture for IT systems. In case new threats are discovered or other requirements require the update of the security schemes the clients are obliged to support the updated scheme.

3.1.1 Trust object exchange

The system establishes initial trust by issuing TSP and TC signing and communication certificates and client credentials. Exchange of trust objects is manual and will be done between TC and EETS TSP security officers.

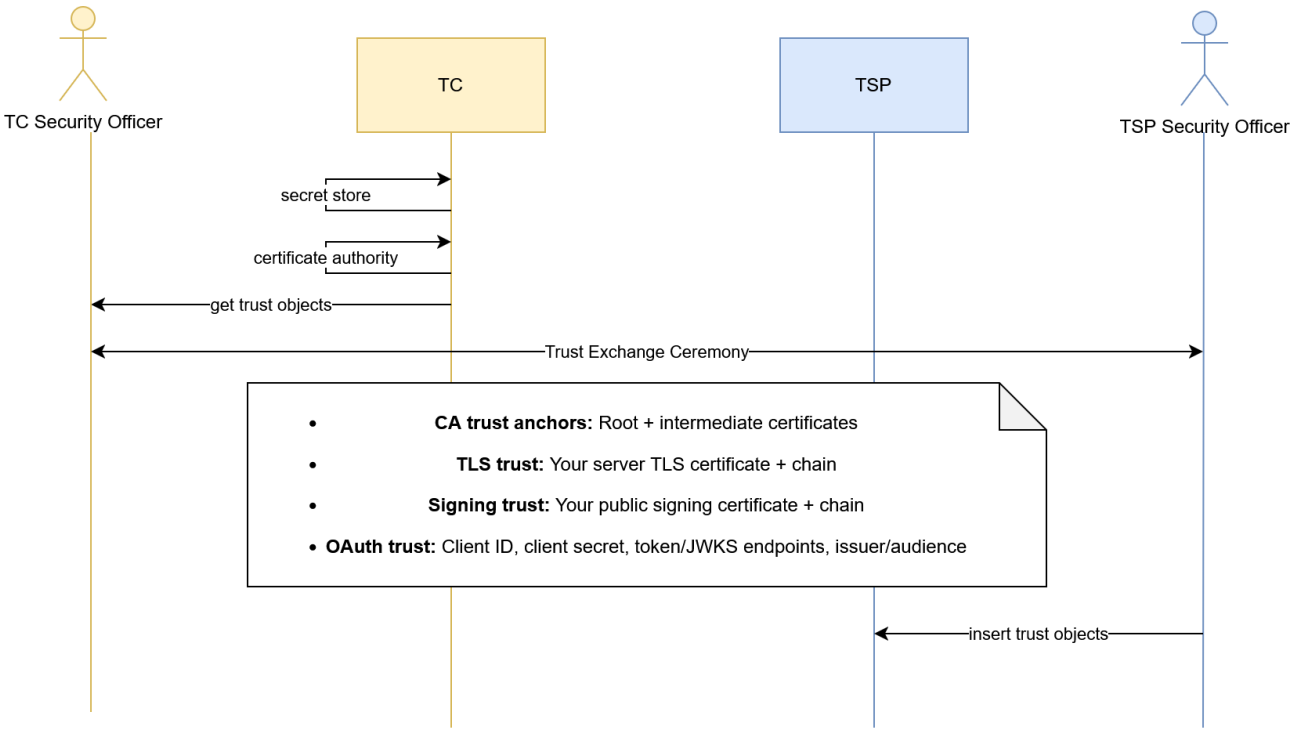

Figure 7 Trust Object Exchange

3.2 Privacy

- The EETS Provider shall ensure compliance with current Directive 95/46/EC (General Data Protection Regulation) and Directive 2002/58/EC concerning the processing of personal data and the protection of privacy and future regulations.
- The EETS Provider shall ensure that the data processed under this contract are hosted within the European Union or any other country having a compliance agreement with the EU.
- The EETS Provider supports the Toll Charger and other Authorities in the fulfillment of Data Protection Legislation.

3.3 General Requirements

The timings and other configurations mentioned are considered configurations and might change according to the needs of the system, for example due to performance considerations. It is expected that TSP solution can adapt to changes required. The system adheres to data minimization principle, meaning it is intended to not collect more data than needed. This requirement may be in contradiction with the ISO 16986 and ISO 12855 standards prescribing mandatory fields, not required by the system. This circumstance is especially prominent in the exchange of vehicle description data. As such the system may receive data from EETS TSP to comply to the standard but may discard them if not needed and in return provide only default values.

4 Configurations and Definitions

This section manages the configuration and definitions applicable to the project.

- END OF DOCUMENT -