



CAMIUNS (NETS) Interface Specification

N-410 - Camiuns (NETS) Interface Specification

Supplement 1 zu Anhang 2 zur Verordnung des BAZG über die technischen und betrieblichen Vorgaben für Anbieter zur Erhebung der leistungsabhängigen Schwerverkehrsabgabe

Version 1.0, 16.02.2024

Table of Contents

1	Document History	5
2	Purpose and audience.....	6
3	NETS Interaction with BAZG	8
3.1	BAZG Interaction possibilities.....	8
3.2	BAZG B2B-Hub-Access Point	9
3.2.1	Design Approach.....	9
3.2.2	Security.....	9
3.3	Partner API v2 (Businesspartner)	9
3.3.1	Open API Specification	9
3.3.2	Endpoints	15
3.3.3	Usage.....	15
3.3.4	Topics	17
3.3.5	Generic Header Attributes	17
3.3.6	Functions.....	18
3.4	BAZG B2B-Hub-Connectivity Setup	20
3.4.1	Registration business partner (by the BAZG)	20
3.4.2	Complete user registration	20
3.5	Access Token	21
3.5.1	Overview	21
3.5.2	Access Token refresh.....	22
3.5.3	Connectivity-Test.....	22
3.6	BAZG Message Definition.....	22
3.6.1	Message Structure	23
3.6.2	Example	23
3.7	Data Integrity	25
3.7.1	Enveloped Signature	26
3.7.2	Certificates	27
3.7.3	Glossary	28
3.8	Transport level / Functional level	29
3.8.1	Example 1: Error at transport level	29
3.8.2	Example 2: Error on functional level.....	29
4	NETS Information Model	31
4.1	TollDeclaration (Message + Interaction Pattern)	31
4.1.1	TollDeclaration Request	31
4.1.2	TollDeclaration Response	32
4.1.3	Acknowledge	34
4.1.4	Interaction Pattern.....	34
4.1.5	Correlation and Message Processing	36
4.2	Notice (Message + Interaction Pattern)	37

4.2.1	Notice Request	37
4.2.2	Acknowledge	38
4.2.3	Interaction Pattern.....	38
4.2.4	Correlation.....	39
5	NETS Conversation Handling	40
5.1	TollDeclarations.....	40
5.1.1	Provider → BAZG	40
5.1.2	BAZG → Provider	41
5.2	Notices.....	42
5.2.1	Provider → BAZG	42
5.2.2	BAZG → Provider	42
6	NETS Messages	44
6.1	Message.....	44
6.2	Message Content Header.....	44
6.3	Message Content Body.....	44
6.4	Notice.....	44
6.5	TollDeclaration (Regular Message)	46
6.6	TollDeclaration (Manual Correction Message)	48
6.6.1	Manual Declaration Reasons.....	51
6.7	TollDeclarationResponse.....	55
6.7.1	ResponseReasons	56
6.7.2	ResponseReasonsText	58
6.8	Acknowledge	58
6.8.1	Issue Codes	58
6.9	Schemas	58
7	NETS Messages-Exchange Scenarios	59
7.1	TollDeclaration (OK)	59
7.1.1	With result ACCEPTED	59
7.2	TollDeclaration (CORRECTION_JOURNEY)	59
7.2.1	Preconditions.....	59
7.2.2	Sequence diagram	60
7.3	TollDeclaration (CORRECTION_TRACK)	60
7.3.1	Preconditions.....	60
7.3.2	Sequence diagram	60
7.4	TollDeclaration (CORRECTION_TRAILER).....	61
7.4.1	Preconditions.....	61
7.4.2	Correction timespan	61
7.4.3	Sequence.....	61
7.5	TollDeclaration (DEADLINE_MISSED)	61
7.5.1	With result REFUSED	61

7.6	TollDeclaration (DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS)	62
7.6.1	With result REFUSED	62
7.7	TollDeclaration (DECLARATION_ID_NOT_UNIQUE)	62
7.7.1	With result REFUSED	62
7.8	TollDeclaration, Notice (ISSUER_ID_UNKNOWN)	63
7.8.1	With result NOK (1002)	63
7.9	TollDeclaration (NO_REGISTRATION_FOR_VIN - vin not found).....	65
7.9.1	With result ACCEPTED ERROR / REFUSED	65
7.10	TollDeclaration, Notice (SIGNATURE_INVALID).....	66
7.10.1	With result NOK (1001)	66
7.11	TollDeclaration, Notice (VALIDATION_ERROR).....	68
7.11.1	TollDeclaration, Notice (VALIDATION_ERROR Schema Level).....	68
7.11.2	TollDeclaration (VALIDATION_ERROR Application Level)	68
7.12	Notice (REGISTRATION)	69
7.12.1	Notice (REGISTRATION_BEGIN).....	69
7.12.2	Notice (REGISTRATION_STATUS).....	69
7.12.3	Notice (REGISTRATION_END).....	70

1 Document History

Version	Date	Changes	Writer	State
1.0	16.02.2024	Version for information of interested NETS Providers	BAZG	final

2 Purpose and audience

This specification describes the technical requirements for providers of the National Electronic Toll Service (NETS) who integrate their IT system with BAZG or the backend system CAMIUNS for the purpose of providing services. The description covers the entire process from obtaining the interface, the underlying information model and the message exchange from a functional and technical point of view.

The technical messages to be used are listed in tabular form.

Message Type	Description	Links
Notice	Notices are used to report the registration or unregistration for vehicles under service	<ul style="list-style-type: none"> • Domain Model <ul style="list-style-type: none"> ◦ Notice • Scenarios <ul style="list-style-type: none"> ◦ Notice (REGISTRATION) ◦ Notice (SERVICE_ERROR)
TollDeclaration	TollDeclarations contain regular or manual declarations for a vehicle under service	<ul style="list-style-type: none"> • Domain Model <ul style="list-style-type: none"> ◦ TollDeclaration (Regular Message) ◦ TollDeclaration (Manual Correction Message) • Scenarios <ul style="list-style-type: none"> ◦ TollDeclaration (OK) ◦ TollDeclaration (DEADLINE MISSED) ◦ TollDeclaration (DECLARATION FROM DIFFERENT PROVIDER EXISTS) ◦ TollDeclaration (DECLARATION ID NOT UNIQUE) ◦ TollDeclaration, Notice (ISSUER ID UNKNOWN) ◦ TollDeclaration (NO REGISTRATION FOR VIN) ◦ TollDeclaration, Notice (SIGNATURE INVALID) ◦ TollDeclaration, Notice (VALIDATION ERROR)
TollDeclarationResponse	Every TollDeclaration will be replied by BAZG with an TollDeclarationResponse.	<ul style="list-style-type: none"> • Domain Model <ul style="list-style-type: none"> ◦ TollDeclarationResponse • Scenarios <ul style="list-style-type: none"> ◦ s. TollDeclaration

Message Type	Description	Links
Acknowledge	Every Message has to be answered. Some Messages arent answered by an TollDeclarationResponse. For those cases, the Acknowledge message type is used.	<ul style="list-style-type: none"> • Domain Model <ul style="list-style-type: none"> ◦ Acknowledge

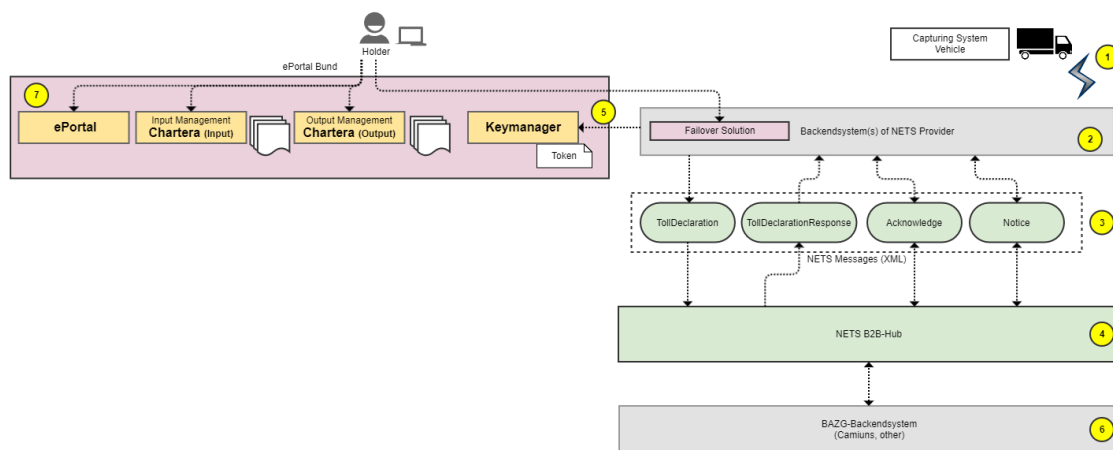
Concrete implementations can be found in section: [Schemas](#). The target audience of this document is the IT departments of the business partners or their architects and developers.

The NETS interface specification will all be all the same for the different groups of NETS providers. Therefore whenever NETS providers are mentioned in the document it is referred to both national NETS provider (NNA) as well as authorized NETS providers (ZNA).

3 NETS Interaction with BAZG

3.1 BAZG Interaction possibilities

For the use of the infrastructure and connection to individual backend systems BAZG provides a common interface. The illustration below shows the principle of how external systems interact with BAZG via "ePortal Bund" and "NETS B2B Hub".



- 1 A vehicle under service capture its journey and transmits the milage over the air to the backend system of a NETS Provider
- 2 The NETS provider's backend system must fulfil the following requirements:
 - 1) It must be able to read the mileage of its contracted vehicles
 - 2) This data must be processed by the BAZG prior to transmission in accordance with the specification
 - 3) A failure solution must be provided for all manual interventions by a vehicle owner. These manual corrections can also result in messages that can be exchanged with the BAZG
 - 4) Communication with the BAZG must take place via the B2B Hub. To be able to communicate successfully with the B2B Hub, a token lifecycle must be implemented using a key manager
- 3 NETS messages are used for data exchange with the BAZG. NETS messages are signed XML files.
- 4 The central system for asynchronous data exchange between a NETS provider and the BAZG is the B2B Hub. It offers a business partner API via which NETS messages can be sent and received.
- 5 The providers backend system(s) obtail the initial API Token via self service portal. For the token refresh, that is necessary to exchange NETS messages, the authorization token can be obtained from a keymanager.
- 6 The messages are processed in the BAZG backend system. This is also where new messages are created, which are provided to a business partner via the B2B Hub.
- 7 A LSVA holder can retrieve and submit various documents (assessment decisions, invoices, etc.) from the BAZG via the ePortal.

3.2 BAZG B2B-Hub-Access Point

3.2.1 Design Approach

Technically, the backend systems of BAZG should be dependent on the IT systems of the business partners:

- No adaptation to backend system BAZG when new providers are added.
- No error handling on backend system BAZG if the providers systems are not available or cannot be reached.

Technically, this is solved with the B2B-Hub, which abstracts a message exchange between the two systems via a clearly defined interface.

Dataflow	Backend System BAZG	Business Partner
incoming BusinessPartner → Backend-System BAZG	Receives messages	Sends messages (push)
outgoing Backend-System BAZG → Business Partner	Publishes messages	Fetches messages (pull)

3.2.2 Security

3.2.2.1 Transport

Transport is protected by means of HTTPS. The exchange of the Nets messages shall be done using the Transport Layer Security, TLS 1.3 protocol.

3.2.2.2 Authentication

The identity of the business partner is guaranteed by means of an Access Token. Such an Access Token is required to access the B2B-Hub. The application of the business partner has to renew the Access Token by its own using credentials (ClientId/ClientSecret).

3.2.2.3 Authorization

The B2B-Hub decides whether the business partner is authorized for the request based on its identity.

3.2.2.4 Integrity

The body of **all** messages (XML) are signed, which can be used to prove (see [BAZG Message Signing](#)):

- that the message was created by this NETS provider
- that the message has not been subsequently modified

3.3 Partner API v2 (Businesspartner)

3.3.1 Open API Specification

[b2bhub-2.0.11-partner-openapi.json](#) master [B2B / b2bhub-api-specs](#)

```
{  
  "openapi": "3.0.1",
```

```

"info": {
  "title": "B2B Hub Partner API",
  "version": "2.0.11"
},
"servers": [
  {
    "url": "https://{url}/api/v2"
  }
],
"tags": [
  {
    "name": "Partner",
    "description": "B2B Hub operations for partners."
  }
],
"paths": {
  "/messages": {
    "get": {
      "tags": [
        "Partner"
      ],
      "summary": "Returns a list of messages for the given BP
(Business Partner), topicName and / or groupId",
      "parameters": [
        {
          "$ref": "#/components/parameters/bpId"
        },
        {
          "$ref": "#/components/parameters/topicName"
        },
        {
          "$ref": "#/components/parameters/groupId"
        },
        {
          "$ref": "#/components/parameters/lastMessageId"
        },
        {
          "$ref": "#/components/parameters/size"
        },
        {
          "$ref": "#/components/parameters/partnerTopic"
        }
      ],
      "responses": {
        "200": {
          "description": "List of messages returned",
          "content": {
            "application/xml": {
              "schema": {
                "$ref": "#/components/schemas/messages"
              }
            }
          }
        },
        "400": {
          "description": "Bad Request"
        },
        "401": {
          "description": "Unauthorized"
        }
      }
    }
  }
}

```

```

        "403": {
            "description": "Forbidden"
        },
        "404": {
            "description": "Not Found"
        },
        "500": {
            "description": "Internal Server Error"
        }
    },
    "x-throttling-tier": "Unlimited"
},
"/messages/{messageId}": {
    "get": {
        "tags": [
            "Partner"
        ],
        "summary": "Get message",
        "parameters": [
            {
                "$ref": "#/components/parameters/messageId"
            },
            {
                "$ref": "#/components/parameters/bpId"
            }
        ],
        "responses": {
            "200": {
                "description": "Message returned",
                "content": {
                    "application/xml": {
                        "schema": {
                            "type": "string"
                        }
                    }
                }
            },
            "400": {
                "description": "Bad Request"
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        },
        "x-throttling-tier": "Unlimited"
    },
    "put": {
        "tags": [
            "Partner"
        ],
        "summary": "Submits a new message",
        "parameters": [
            {

```

```

        "$ref": "#/components/parameters/messageId"
      },
      {
        "$ref": "#/components/parameters/bpId"
      },
      {
        "$ref": "#/components/parameters/messageType"
      }
    ],
    "requestBody": {
      "description": "XML Content",
      "content": {
        "application/xml": {
          "schema": {
            "type": "string"
          }
        }
      }
    },
    "required": true
  },
  "responses": {
    "201": {
      "description": "Created"
    },
    "400": {
      "description": "Bad Request"
    },
    "401": {
      "description": "Unauthorized"
    },
    "403": {
      "description": "Forbidden"
    },
    "500": {
      "description": "Internal Server Error"
    }
  },
  "x-throttling-tier": "Unlimited",
  "x-codegen-request-body-name": "body"
}
},
"/messages/{messageId}/next": {
  "get": {
    "tags": [
      "Partner"
    ],
    "summary": "Get next message",
    "parameters": [
      {
        "$ref": "#/components/parameters/messageId"
      },
      {
        "$ref": "#/components/parameters/bpId"
      },
      {
        "$ref": "#/components/parameters/topicName"
      },
      {
        "$ref": "#/components/parameters/partnerTopic"
      }
    ]
  }
}

```

```

    ],
    "responses": {
      "200": {
        "description": "Message returned",
        "headers": {
          "messageId": {
            "description": "messageId of the returned
message",
            "schema": {
              "type": "string",
              "format": "uuid"
            }
          }
        },
        "content": {
          "application/xml": {
            "schema": {
              "type": "string"
            }
          }
        }
      },
      "400": {
        "description": "Bad Request"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      },
      "500": {
        "description": "Internal Server Error"
      }
    },
    "x-throttling-tier": "Unlimited"
  }
},
"components": {
  "schemas": {
    "messages": {
      "type": "array",
      "items": {
        "$ref": "#/components/schemas/message"
      }
    },
    "message": {
      "title": "message",
      "required": [
        "messageType",
        "bpId",
        "messageId"
      ],
      "type": "object",
      "properties": {
        "messageId": {

```

```

        "type": "string",
        "format": "uuid",
        "description": "The unique identifier of the message",
        "example": "c4d4c852-c134-41c2-96b2-382ebcdac5f9"
    },
    "messageType": {
        "type": "string",
        "description": "Business type definition of the message
body",
        "example": "XX123"
    }
},
"description": "Message for the B2BHub"
}
},
"parameters": {
    "messageId": {
        "name": "messageId",
        "in": "path",
        "description": "Unique message identification as UUID: cc7d5097-
4d3f-4fff-af91-fd3680199642",
        "required": true,
        "schema": {
            "type": "string",
            "format": "uuid"
        }
    },
    "bpId": {
        "name": "bpId",
        "in": "header",
        "description": "Partner identification",
        "required": true,
        "schema": {
            "type": "string"
        }
    },
    "messageType": {
        "name": "messageType",
        "in": "header",
        "description": "Business type definition of the message body",
        "required": true,
        "schema": {
            "type": "string"
        }
    },
    "topicName": {
        "name": "topicName",
        "in": "query",
        "description": "Get only messages from given topicName",
        "required": false,
        "schema": {
            "type": "string"
        }
    },
    "groupId": {
        "name": "groupId",
        "in": "query",
        "description": "Get only messages with given groupId",
        "required": false,
        "schema": {

```

```
        "type": "string"
      }
    },
    "lastMessageId": {
      "name": "lastMessageId",
      "in": "query",
      "description": "Get only messages which were published after the
given messageId",
      "required": false,
      "schema": {
        "type": "string",
        "format": "uuid"
      }
    },
    "size": {
      "name": "size",
      "in": "query",
      "description": "Number of messages returned",
      "required": false,
      "schema": {
        "type": "integer",
        "default": 1000,
        "minimum": 1,
        "maximum": 1000
      }
    },
    "partnerTopic": {
      "name": "partnerTopic",
      "in": "header",
      "description": "Partner Topic",
      "required": false,
      "schema": {
        "type": "string"
      }
    }
  }
}
}
```

3.3.2 Endpoints

URL Production Environment	https://nets-b2bhub.bazg.admin.ch/declaration/api/
URL Acceptance Environment	https://abn-nets-b2bhub.bazg.admin.ch/declaration/api/

3.3.3 Usage

Endpoint	Method	Usage
/messages	GET	Returns a list of NETS messages for the specific business partner by bpId, topicName and groupId. The response (XML) contains a list of messageIds with the messageType of each NETS message.

Endpoint	Method	Usage
/messages/{messageId}	GET	Returns the NETS message for the specified business partner by bpId and message by messageId. The response (XML) fulfils the corresponding XSD according to the specification of the NETS message.
/messages/{messageId}/next	GET	Returns a message for the specified business partner by bpId and/or topicName, which was published chronologically after the message with the specified messageId and is used to iterate over messages in chronological order. The response (XML) fulfils the corresponding XSD in accordance with the NETS message specification
/messages/{messageId}	PUT	Used to transmit a new message. The request body must be an XML that fulfils the corresponding XSD according to the NETS message specification

3.3.3.1 Message Identifier

The messageId is used to uniquely identify a message that is transmitted via the B2B Hub. This identification is set by the source system and corresponds to a UUID. It must fulfil the following convention:

[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

If a messageId is sent several times, the previously sent document is overwritten each time.

3.3.3.2 Usage Variants

A business partner can collect messages from the B2B Hub in two ways:

1. Retrieve a list of messages with /messages and then retrieve each individual message. As soon as all messages have been retrieved, retrieve the next messages with /messages?lastMessageId.
2. Fetch first message /messages?size=1 and then only iterate over all messages with /next until a 404 is reached.

Variant	List & single retrieval	Iteration with /next
Initial request	/messages?size=2 Response: <messages> <messageId>1</messageId> <messageId>2</messageId> </messages>	/messages?size=1 Response: <messages> <messageId>1</messageId> </messages>
Frist message	/messages/1 Response: Header: messageId:1 Body: <xmlContent>	/messages/1 Response: Header: messageId:1 Body: <xmlContent>

Variant	List & single retrieval	Iteration with /next
Next message	/messages/2 Response: Header: messageId:2 Body: <xmlContent>	/messages/1/next Response: Header: messageId:2 Body: <xmlContent>
Next message	/messages?size=2&lastMessageId=2 Response: <messages> <messageId>3</messageId> <messageId>4</messageId> </messages> /messages/3 Response: Header: messageId:3 Body: <xmlContent>	/messages/2/next Response: Header: messageId:3 Body: <xmlContent>

3.3.4 Topics

By distributing the messages to different topics, we enable the counterpart to allow different software components to access "their" messages directly without the need for forwarding.

The following topics are currently implemented for the traffic duties:

Business Object	Topic	Description
Notice TollDeclaration (Regular Message) TollDeclaration (Manual Correction Message) TollDeclarationResponse Acknowledge	nets-tolldeclaration	all NETS messages share the same topic

Topics are only available for outgoing messages.

3.3.5 Generic Header Attributes

This header attributes should be part of every API Call.

Name	Required	Description	Example Value
bpId	yes	The Business Partner Id. Must match the Business Partner in the token.	1234567891
Authorization	yes	The token goes here	Bearer eyJ0eX...

3.3.6 Functions

3.3.6.1 Fetch message(s)

The BAZG provides 3 different API endpoints for querying and retrieving messages.

3.3.6.1.1 List message(s)

Verb	GET			
Endpoint	/messages/			
Description	Returns a list of messages for the specified business partner, topicName and / or groupId			
Direction	BAZG → BP			
Header Attributes	Generic header attributes			
Path Parameter				
Query Parameter	Name	Required	Description	Example Value
	topicName	no	the name of the topic	nets-tolldeclaration
	lastMessageId	no	Filter to only see messages after a previous MessageId	UUID
	size	no	Number of messages which are returned: <ul style="list-style-type: none"> • default: 1000 • minium: 1 • maxium: 1000 	1000

3.3.6.1.2 Fetch single message

Verb	GET			
Endpoint	/messages/{messageId}			
Description	Returns the NETS message for the specified business partner by bpId and message by messageId.			
Direction	BAZG → BP			
Header Attributes	Generic header attributes			
Path Parameter	Name	Type	Description	Example Value
	messageId	UUID	s. Message Identifier	12345678-1234-1234-1234-123456789012
Query Parameter				

3.3.6.1.3 Fetch next message

Verb	GET		
Endpoint	/messages/{messageId}/next		
Description	Returns a message for the specified business partner and/or topicName, which was published chronologically according to the given MessageID.		
Direction	BAZG→ BP		
Header Attributes	Generic header attributes		
Path Parameter	Name	Type	Description
	messageId	UUID	s. Message Identifier
Example Value	12345678-1234-1234-1234-123456789012		
Query Parameter			

3.3.6.2 Submit message

Verb	PUT		
Endpoint	/messages/{messageId}		
Description	Transmits a new message		
Direction	BP → BAZG		
Header Attributes	Generic header attributes +		
	Name	Required	Description
	content-type	yes	the content type has to be application/xml
messageType	yes	The messageType is used for the unique assignment of a sent message. This is set by the sender of the message. <ul style="list-style-type: none"> • <ul style="list-style-type: none"> ○ nets-acknowledge for <u>Acknowledge</u> messages ○ nets-notice for <u>Notice</u> messages ○ nets-manualtolldclaration for <u>TollDeclaration (Manual Correction Message)</u> messages ○ nets-regulartolldclaration for <u>TollDeclaration</u> 	nets-acknowledge

				(Regular Message) messages
Path Parameter	Name	Type	Description	Example Value
	messageId	UUID	s. Message Identifier	12345678-1234-1234-1234- 123456789012
Query Parameter				
Body	The HTTP body must be an XML (content type application/xml) The body is validated for XML conformity by the B2B Hub.			

In case the Message is successfully received and stored by the BAZG, the request will be answered with HTTP (200).

Remarks:

- This operation is idempotent. This means in case of timeout the NETS provider can safely repeat the operation, using exactly the same messageId, until the operation is successful.
- Replacing of a messages is not supported. BAZG will always use the first message successfully received for the messageId specified.

3.4 BAZG B2B-Hub-Connectivity Setup

Each counterpart, which communicates via B2B with LSVA III, must setup his system to operate with Camiuns.

3.4.1 Registration business partner (by the BAZG)

Based on the company details, the corresponding business partner is created by BAZG. The onboarding code is sent by e-mail and by post to the contact person of the company.

3.4.2 Complete user registration

Step	Title	Description
1	Register User	<ol style="list-style-type: none"> 1. Call the ePortal: https://eportal-a.admin.ch/ (Acceptance Environment) 2. Button "Jetzt Benutzerkonto erstellen" 3. Create new account incl. two-factor authentication in the system eIAM 1).
2	Redeem Onboarding Code	<ol style="list-style-type: none"> 1. Entry header navigation "Geschäftspartner verwalten" 2. Link "Zur EZV Geschäftspartnerverwaltung" 3. Link "Onboarding-Code einlösen" and follow instructions.
3	Subscribe API	https://abn-selfservice-ezv.api.admin.ch/api-selfservice/apis
4	Deposit generated tokens	The token generated in the ePortal is stored in the business partner's business software solution (customer system).

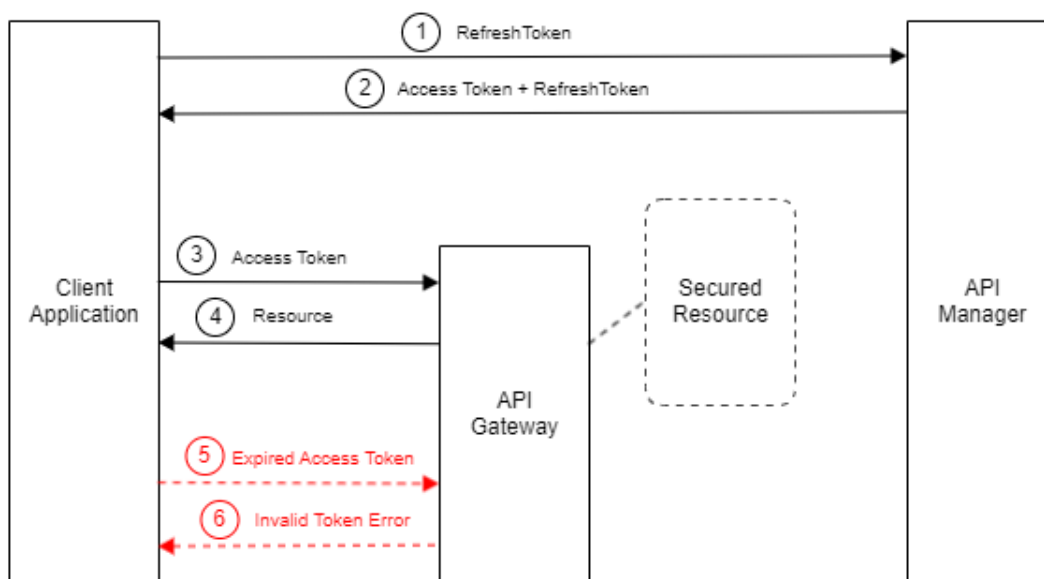
1) eIAM is the central access and authorization system of the federal administration for web applications, and native mobile apps. Therefore a registration is needed to get access to the ePortal.

3.5 Access Token

3.5.1 Overview

A valid access token is required to call up a protected API. An access token is only valid for a certain period of time. A new access token can be requested using a refresh token. A new refresh token is also created in the process. This can be used to request another Access Token together with a Refresh Token.

To request a new Access Token, a Client ID and Client Secret are required in addition to the Refresh Token.



- 1 The client application requests a new access token from the API Manager with a refresh token.
- 2 If the refresh token is valid, the API Manager issues a new access token and a new refresh token and sends both to the client application.
- 3 The client application calls the API for a secured resource on the API gateway with a valid access token.
- 4 If the access token allows access to the secured resource, the API gateway forwards the request and returns the response.
- 5 The client application calls the API for a secured resource with an expired access token.
- 6 The API gateway rejects the request with an "Invalid Token Error".

3.5.2 Access Token refresh

To avoid a manual creation of a new set of tokens every 24 hours via API Selfcare, the automatic Access Token Refresh shall be implemented as part of the registration process on the customer system.

URL of the refresh-API from the acceptance-environment * <https://keymanager.api.admin.ch/token/v2>

* is identical for all environments

3.5.2.1 Example request (with curl):

```
curl -k -u <clientId>:<clientSecret> -d  
"grant_type=refresh_token&refresh_token=<refresh-token>"  
https://keymanager.api.admin.ch/token/v2
```

The response is a new token, which can be used for transmission afterwards:

```
{"access_token": "{Bearer_token_goes_here}", "refresh_token": "{Refresh_token_goes_here}"  
", "scope": "default", "token_type": "Bearer", "expires_in": 86400}
```

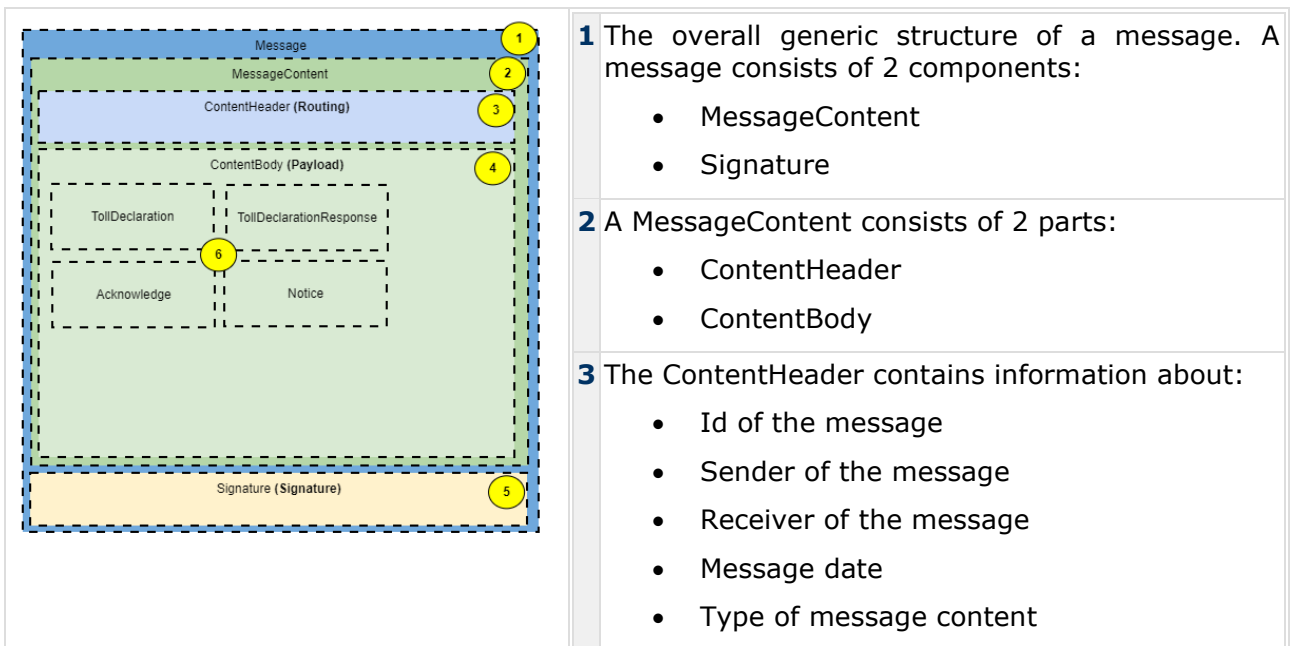
If the bearer token expires, a new bearer token can be generated from the response with the refresh token.

3.5.3 Connectivity-Test

The connectivity test consists of sending a message from the customer system to the BAZG. When the message arrives at the BAZG, the connectivity test is considered passed.

3.6 BAZG Message Definition

A message has a generic structure and allows the definition of all information necessary for the exchange. During message exchange the message as described hereafter is enclosed in the datafield "body". See also [BAZG B2B-Hub-Access Point](#).



-
- | | |
|--|---|
| | 4 The ContentBody contains the specific message (defined by a specific MessageType). |
| | 5 The signature ensures the authenticity of the MessageContent (s. BAZG Message Signing). |
| | 6 A specific business context is defined by it's information model. Different message types can be used to describe the model. |

3.6.1 Message Structure

The structure and characteristics are listed in chapter [NETS Messages](#).

3.6.2 Example

3.6.2.1 Nets Notice

Below an example of a complete message (header + content) of a message of type notice.

```

<?xml version="1.0" encoding="UTF-8"?>
<message xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
  <messageContent>
    <contentHeader>
      <messageId>7a5a323c-6ec6-4889-85af-05cf27351d99</messageId>
      <messageDateTime>2024-01-29T13:11:01.316+01:00</messageDateTime>
      <informationSenderId>
        <issuerId>1234567891</issuerId>
      </informationSenderId>
      <informationRecipientId>
        <issuerId>1000006447</issuerId>
      </informationRecipientId>
    </contentHeader>
    <contentBody>
      <notice>
        <noticeId>1706530261316</noticeId>
        <noticeType>REGISTRATION_BEGIN</noticeType>
        <dateTimeWhenRecorded>2024-01-
29T13:11:01.316+01:00</dateTimeWhenRecorded>
        <recordedBy>PROVIDER</recordedBy>
        <registrationBegin>
          <vin>XLRASH4300G232849</vin>
          <tollDomain>CH</tollDomain>
          <effectiveDate>2024-01-29</effectiveDate>
        </registrationBegin>
      </notice>
    </contentBody>
  </messageContent>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-
c14n11"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha256"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <ds:DigestValue>b3HAFHg5pQoRHx4jybphBdw0sHKsX7oSG00Q4bK8+LE=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
Za51k40MxLTJ5o+57Gp9XxWVKwlLSB61iJe0sxBTg7v1/iFzHQH0qhv0H7bOQQoFv6cre3AsS3FK
IATL4LMZHQcVRxRjyahEVBRC90sAeX4uGKqX2T6mXlid9A2Z2WhDqPHzx1n8sNatwQefPc+zT+nc
ADDR9t6rROIHZG5zPM8Ia3F4h6a3MclTZxZvjSaCHylL/eDSEYp5IkcrvJA8oNvm6EOxXn79ygqA
8iaaVtFMwQOMucxBrJvuNZ6s9c+9NeoAZZDH02Np0gwrLlR5/A1At3iRitF2NEI9jiWE68tv18S5
G3enecgvQqgCP45ihQn2AMu2YNrGhDXH1DZdMw==
</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
MIIDrDCCApSgAwIBAgIIZIdmZRMt0zswDQYJKoZIhvcNAQELBQAwfTElMAkGA1UEBhMCUEwGzAZ
BgNVBAGTElphY2hvZG5pb3BvbW9yc2tpZTERMA8GA1UEBxMIU3pjemVjaW4xFTATBgNVBAoTDHBy

```



```

b3ZpZGVyIG9yZzEwMBQGA1UECXMNcHJvdm1kZXIgdGVzdDEPMA0GA1UEAxMGdGVzdCAxMCAxDTIz
MDgwMTEwNTQwMFoYDzIxMjMwODAxMTE1NDAwWjB9MQswCQYDVQQGEwJQTDEbMBkGA1UECBMSWmFj
aG9kbmlvcG9tb3Jza211MREwDwYDVQQHEWhTemN6ZWNPbjEVMBMGA1UEChMMcHJvdm1kZXIgb3Jn
MRYwFAYDVQQLEw1wcm92aWR1ciB0ZXNOMQ8wDQYDVQQDEwZ0ZXN0IDEwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCQoYI+Euq2P/y5g5pIK7LCoMknNCeneUaf7YDrhBnJBu6R6NrVOUMx
MnTj jrUF074a/4LKH6hSE6qJE9M9+gNxdfMntfKrvXFd4CnvUmFyfXuDCs+Qbysj9No/Jbah20ku
gcymkdz2rgS/6r2kaGG7QpUVH61FMHPqkUwUi24goBxwmghs5Palzn87GJcHdFmvQjQp8W2eM2/
x3mh63NRlRgOQs1RmLHmgZMIQ/9edHYi4Ji9NReEEUuy4FcqA3ro71hndKbv61BRMMKgWreeEfe
Nlq3wDhGe7raceY06BRgeXx0dPDR4fs+bjFVkeVaZF6KmJo2fx5I9WWd8XtZAgMBAAGjLjAsMBOG
A1UdDgQWBBQd5DCgbKt0WUX39jEKcTsXacsYGjALBgNVHQ8EBAMCB4AwDQYJKoZIhvcNAQELBQAD
ggEBAG5vQJI075+E7AkvHAqmo/+p+vfwDQzii1NBbK8rmdn06nyFZWLHHJ2Vevj8z4psUJhf/u3
xVuQEmZ67xpRPLE+0hrW9rUKSTvAVOSiPGI5Ten5wYqaljr6eAgPqKw839YJsJcLhWHDMVkJZMyLF
yhOjXVhFa/i/TBSp5BtY4Ej2ZhRVWp8Gd2Sf4rpoTgNlzxSPY1wxWnwPv1jW52LFvWr8yuQ6y1H+
iwF19/7LJs7WdxnbVXmqb9CdB/75bqaTKTiH0R6H7/HYVZULW9FGr12A3bilkr6pk/MwsBBNVw3I
AU1/ZZK1c763wb9Apd5Ij/YUctpjpIjUWYwca9IkG/k=
</ds:X509Certificate>
  </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</message>

```

3.7 Data Integrity

All messages exchanged between the business partner and BAZG shall be signed. BAZG will archive this message including signature in order to be able to prove:

- that the message in the archive was actually created by a specific provider (information origin)
- that the message in the archive has not been subsequently changed (information integrity)

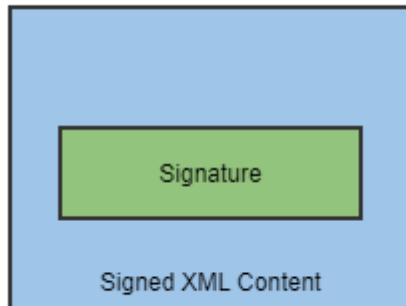
The digital signatures of the NETS messages shall be according to the W3C standard "XML Signature Syntax and Processing Version 1.1" (W3C Recommendation 11 April 2013, <https://www.w3.org/TR/2013/REC-xmldsig-core1-20130411>). From that standard the following options shall be used:

- Canonicalization of the XML document, Identifier for canonical XML 1.0 (omits comments): <http://www.w3.org/2006/12/xml-c14n11>
- Signature with envelope, Identifier: <http://www.w3.org/2000/09/xmldsig#enveloped-signature>
- For the <KeyInfo> element, the option <X509Data> (Identifier <http://www.w3.org/2000/09/xmldsig#X509Data>) with exactly one attribute of type <X509SubjectName> and <X509Certificate>

-
- Signature algorithm PKCS#1 v1.5 with digest sha256 (at least), identifier: <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>

3.7.1 Enveloped Signature

Means that a <signature> element will be added to the part of the message to be signed. We will sign the whole message, which will embed the signature into the message.



The signed message has the following structure:

```

<?xml version="1.0" encoding="UTF-8"?>
<Message>
  <MessageContent>
    <ContentHeader/>
    <ContentBody/>
  </MessageContent>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-
c14n11"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-
more#rsa-sha512"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
        </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>ubGfhen7JeSRL0BYVMace6Do6sBuzx/B6ZcW9mQVdoQ=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
        ...
      </ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
            ...
          </ds:X509Certificate>
          <ds:X509SubjectName>...</ds:X509SubjectName>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
  </Message>

```

3.7.2 Certificates

3.7.2.1 General

The used certificates shall be according X.509. The length of the RSA keys in the X.509 certificates shall be at least 2048 bits.

3.7.2.2 BAZG

The currently used BAZG XML document signature certificate shall be published on the BAZG webpage,

Documents

Insert Links to online available documents

3.7.2.3 NETS provider

The NETS provider shall provide the BAZG with the necessary information to validate the message signatures.

This can either be in the form of a public key or a certificate of his choice.

The NETS provider is responsible for notifying the BAZG if the security of one of the certificates can no longer be guaranteed or if he wants to renew his signing key.

3.7.3 Glossary

Parameter	Applicable Specification	Description
CanonicalizationMethod	http://www.w3.org/2006/12/xml-c14n11	<p>The XML syntax allows a number of options (e.g. single instead of double quotes for attribute values, the order of attributes in a tag, places with empty (or blank) spaces are irrelevant) which can result in a physically different but logically identical document. The purpose of normalization is to define an algorithm that reliably and repeatably transforms any XML document into its canonical (simplest, normalized) form. If the same algorithm is applied to 2 physically different documents, they can then be compared in logical form.</p> <p>The CanonicalizationMethod is applied to the SignedInfo element, not to the data. The data must be transformed with the algorithms listed under Transform before the digest can be calculated.</p>
Reference		Points to the resources (data) that must be signed. In our case, the signature is not restricted to individual resources, so the reference URI is empty.
Transformation	http://www.w3.org/2000/09/xmlsig#enveloped-signature http://www.w3.org/2006/12/xml-c14n11	<p>The following transformation algorithms are applied to the data before the digest is calculated:</p> <ul style="list-style-type: none"> • Canonicalisation • Enveloped Signature
DigestMethod	http://www.w3.org/2001/04/xmlenc#sha256	For performance reasons, public-key algorithms usually do not sign the entire document, but only a hash of the entire document. This summary is called a digest. The digest is applied via the URI listed under References. The digest is Base-64 encoded. This digest is then integrated into the SignedInfo element to be signed.
DigestValue		The calculated digest (the hash of the entire data).
SignatureMethod	http://www.w3.org/2001/04/xmlsig-more#rsa-sha512	Algorithm which is used to sign the normalized SignedInfo.
SignatureValue		The base-64 encoded result of signing the entire normalized SignedInfo element.
KeyInfo	http://www.w3.org/TR/xmlsig-core/#sec-X509Data	<p>We use X.509 certificates. In the signature X509SubjectName and X509Certificate are expected.</p> <pre><X509Data> <X509SubjectName>CN=My Name,O=Test Certificates Inc.,C=US</X509SubjectName> <X509Certificate>MIIB9zCCAWCgAwIBAgIERZwdk zANBgkqhki...</X509Certificate> </X509Data></pre>

In order for a NETS provider to be able to provide its services, it must connect its IT infrastructure to BAZG systems. The message exchange is based on the specifications defined by BAZG / LSVA III.

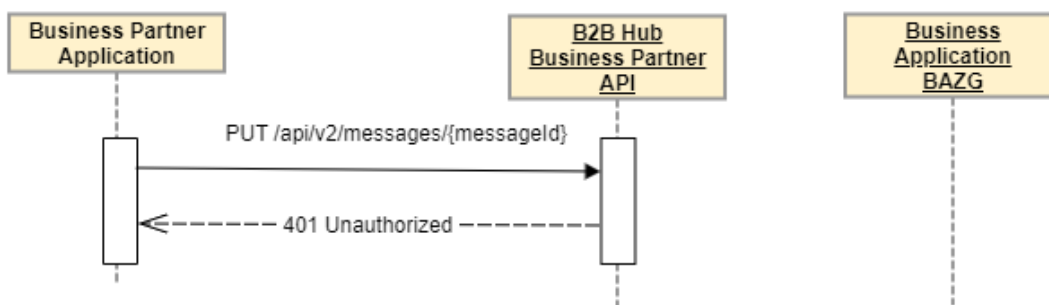
3.8 Transport level / Functional level

Due to the asynchronous structure of the [BAZG B2B-Hub-Access Point](#), a distinction must be made between the (HTTP) transport level and the so-called "functional" level:

- If the business partner transmits a message, he receives a response immediately after transmission (transport layer).
- The message is then processed by the BAZG system and generates a response (functional level).

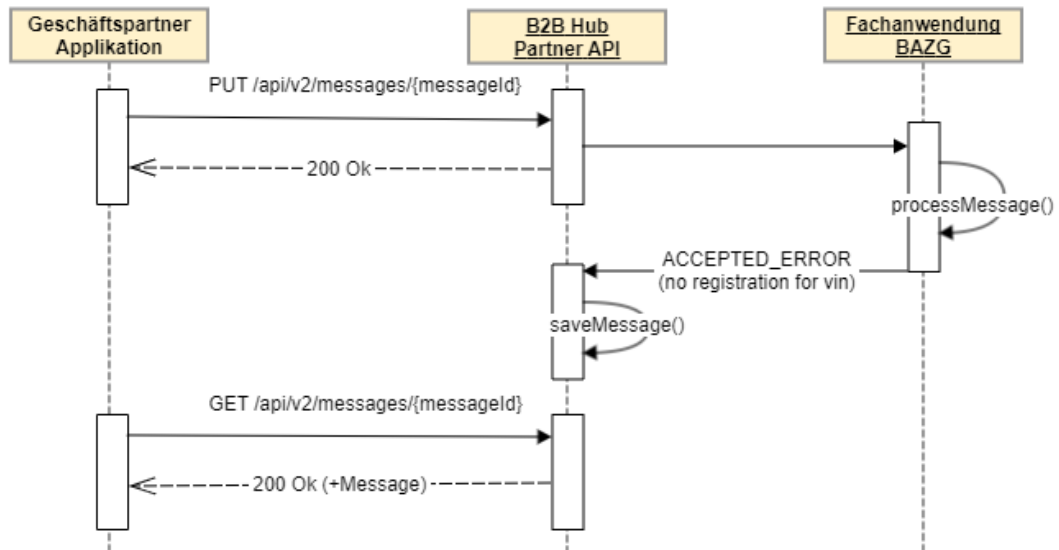
3.8.1 Example 1: Error at transport level

A LSVA III provider transmits a message and forgets to add the bearer token to the request. When the message is transmitted, an HTTP 401 (unauthorized) error is received at the transport level.



3.8.2 Example 2: Error on functional level

Business partner transmits a message (ex. TollDeclaration) which is accepted by the B2B hub. At the transport level, the HVC provider receives an ok (HTTP 200). The message is then processed in the specialized application at the BAZG, where in this specific example it is determined that there is no registration for the vehicle yet. This error on the functional level is reported back to the business partner with a specific message.



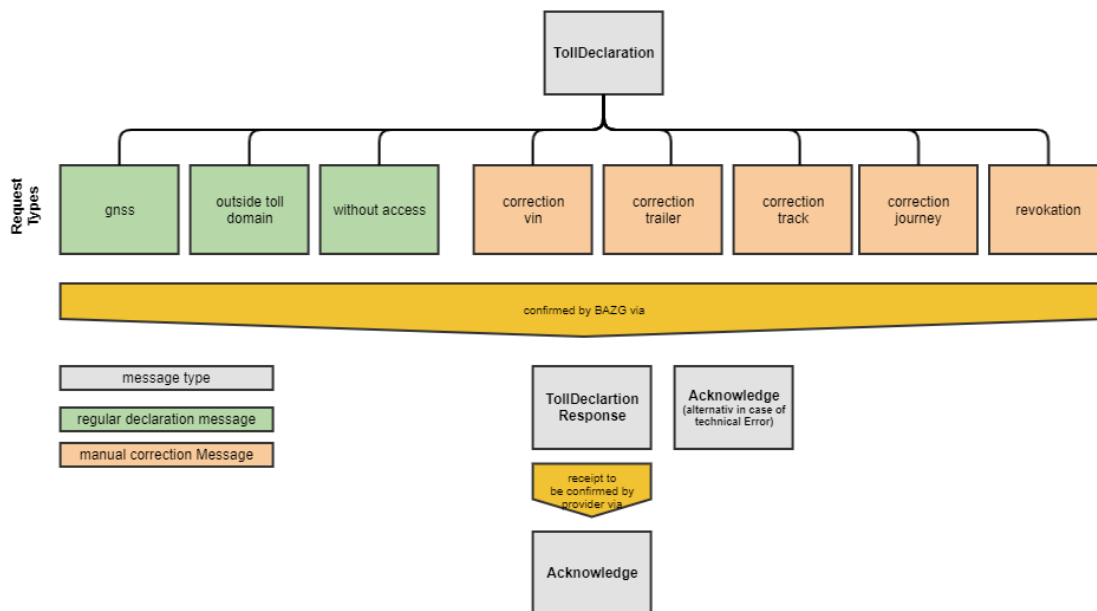
4 NETS Information Model

The information model describes the message structure and message exchange in the context of NETS. Two types of messages and exchange patterns are provided to interact with BAZG / CAMIUNS.

- **TollDeclaration** together with TollDeclarationResponse and Acknowledge for the purpose of daily registration of mileage and the associated levy-determining parameters.
- **Notice** for a more general message exchange used for ex. during registration and deregistration process.
- **Acknowledge** for the confirmation of a message

4.1 TollDeclaration (Message + Interaction Pattern)

4.1.1 TollDeclaration Request



4.1.1.1 TollDeclaration Request Types (Regular Declaration)

The regular declaration message is vehicle-related, and relates to a specific period (=1 day). It is used to deliver driving data in the form of gns information or when no gns data is available or the vehicle is outside toll domain a corresponding status information for the vehicle. Therefore, following request message types exist.

Request Type	Description
gns	Contains the driving data in the form of gns information including and where available the corresponding trailer information.

Request Type	Description
outside domain	toll The vehicle was completely outside the toll domain during the relevant period.
without access	The vehicle's driving data could not be retrieved for the relevant period.

4.1.1.2 TollDeclaration Request Types (Manual Correction)

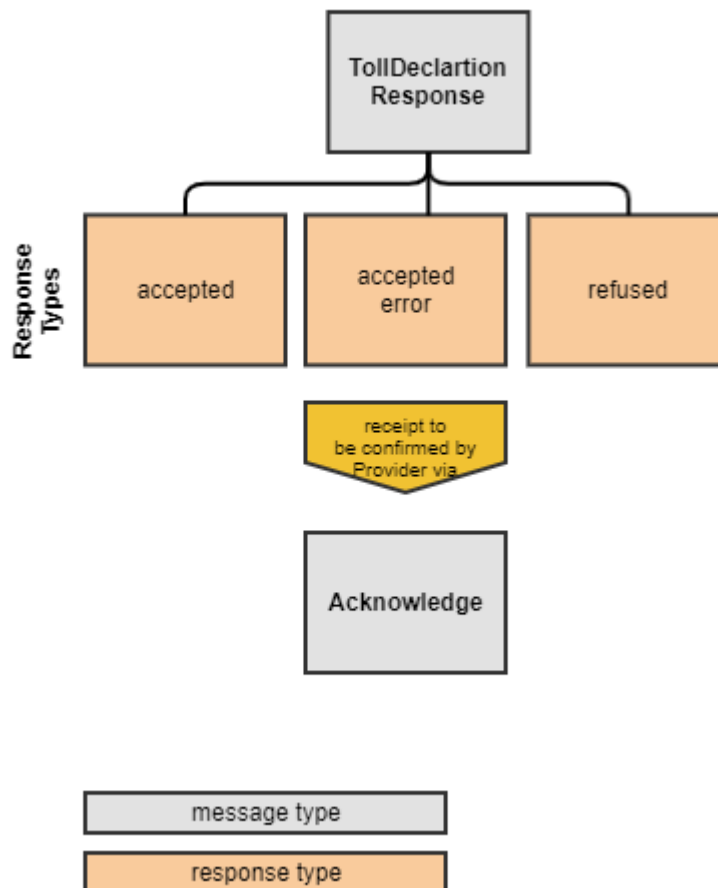
A manual correction references to a previously sent regular declaration or parts of it or in some cases to activated manual corrections. For detailed information of the correlation between regular declaration message and manual correction message see: "Correlation and processing activated messages".

The following types of manual corrections are possible:

Request Type	Description
correction vin	Corrects the vin of regular declaration(s) and if existing manual correction(s).
correction trailer	Corrects the declared trailer information in the provided timeframe not altering declared distance.
correction track	Corrects a regular declaration by adding a missing track providing distance and trailer information.
correction journey	Via a regular declaration driving data was registered where there was no journey subject to tax (eg. in cases when vehicle gets transported by a truck, on-board unit has been moved outside the vehicle, ...). Additionally proactive declaration of days without journey subject to tax can be declared using correction journey.
revokation	Revokes a submitted manual correction.

4.1.2 TollDeclaration Response

A successfully received tollDeclaration is confirmed by a tollDeclarationResponse. The same response is used for all message types.

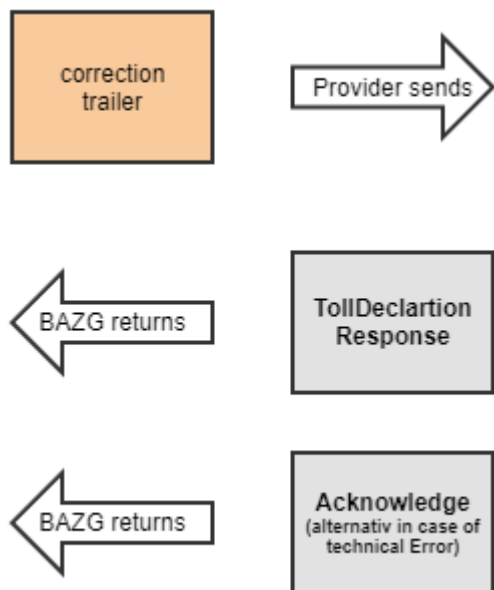


4.1.2.1 TollDeclarationResponse Response Types

Response Types	Description
accepted	Declaration has been accepted by BAZG.
accepted error	Declaration has been accepted by BAZG with a signaled error. Action must be taken by provider.
refused	Declaration has been refused by BAZG. The message will not be processed for given reason(s). Resending after correction is possible.

4.1.2.2 Technical errors

Due to the asynchronous nature of the B2B Hub message exchange, every message (whether from the NETS provider or the BAZG) must be confirmed. For messages of type Notice, an acknowledgement is confirmed in the form of an Acknowledge message. The message exchange for TollDeclarations is much more complex, so a separate message type is available named TollDeclarationResponse. In order to avoid having to implement generic errors such as xml schema violations or problems with the signature twice, no TollDeclarationResponse is reported back in the event of these technical errors, but an Acknowledge with ackCode NOT_OK is sent back.



This means that once a TollDeclaration has been transmitted, the provider's backend system must be able to handle both messages of type Acknowledge and messages of type TollDeclarationResponse.

4.1.3 Acknowledge

4.1.3.1 BAZG → Provider

In cases where a sent message cannot be received and processed by CAMIUNS a response on message level shows the reason for rejection. See: [Acknowledge](#).

4.1.3.2 Provider → BAZG

Each tollDeclaration message received by the provider needs to be confirmed by a corresponding acknowledge. See: [Acknowledge](#).

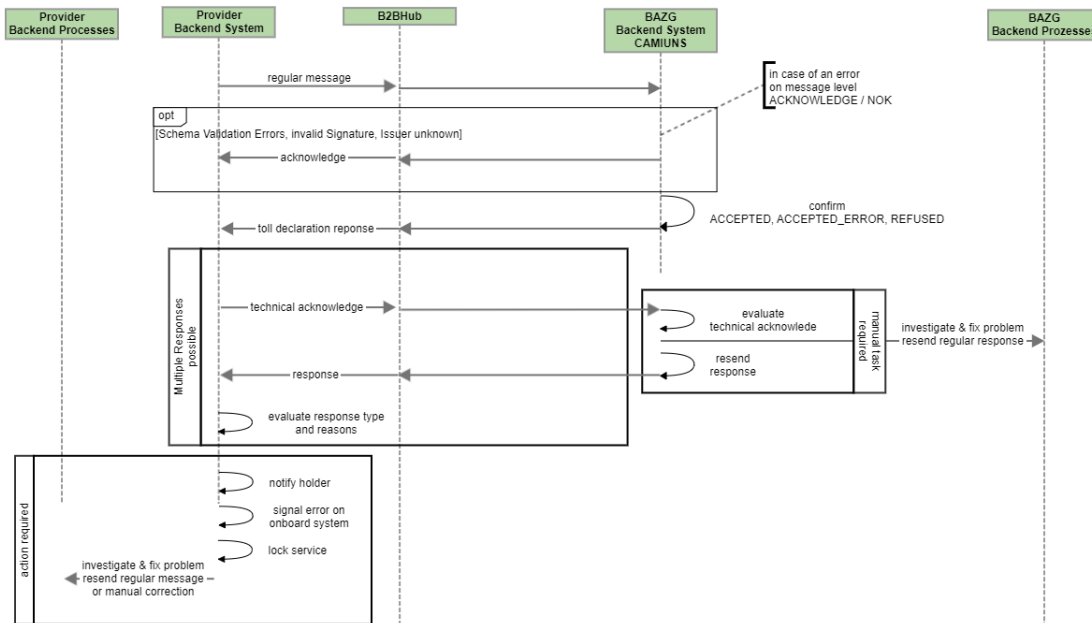
An acknowledge on business level is given usually within a few minutes to an hour. In cases of system downtime it might be delayed up to 24h. If an acknowledge is not provided within 24h, the message must be resent with its original content after 24h / 48h. If an acknowledge is still pending after 48 hours BAZG has to be contacted.

4.1.4 Interaction Pattern

The message exchange for a regular declaration message and a manual correction follow a similar interaction pattern.

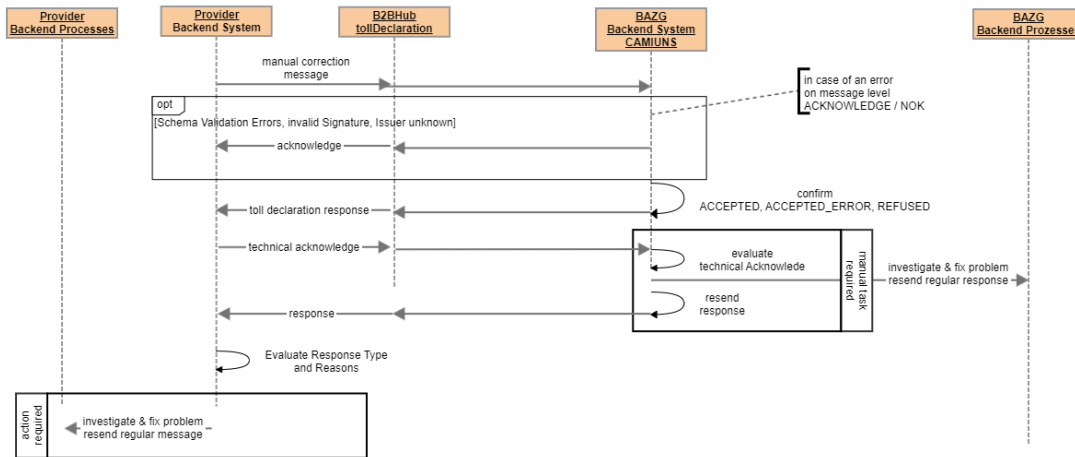
4.1.4.1 Regular Declaration Message

In case of a declaration via regular declaration message the following interaction pattern has to be followed.



4.1.4.2 Manual Correction Message

In case of a declaration via manual correction message the following interaction pattern has to be followed.



4.1.4.3 Message Handling Rules

Beside the interaction pattern described above following rules apply:

4.1.4.3.1 for regular declaration messages:

- Regular declaration messages are identified by vin and declaration period.
- For each vehicle and declaration period at least one regular declaration has to be sent.
- The type can be either:

-
- gnss (shall also be used in cases where there was no journey but provider could access the onboard system).
 - outside toll domain
 - without access.
 - Declarations have to be done on a daily basis and
 - they may be continuous.
 - Submitted tracks shall not overlap.
 - Submissions of one day by means of multiple registrations will be merged in CAMIUNS.
 - Submission is due on the next day following the declaration period but can be sent up to 10 days after the declaration period. Any declaration sent after that date will be refused. The date and time of receipt of the notification by the BAZG is decisive.
 - Each regular declaration accepted by BAZG will be legally binding after the 10 days.

4.1.4.3.2 for manual correction messages:

- Manual correction messages are identified by vin and declaration period
- Manual correction targets a whole declarationPeriod or part of it
- Manual correction can be sent as soon as data is available
- Manual correction are accepted up to 10 days after the relevant declaration period. The date and time of receipt of the notification by the BAZG is decisive.
- Manual correction shall be delivered at the latest the following day after they have been filed in the system of the provider.
- Each manual correction message accepted by BAZG will be legally binding 10 days after the concerned declaration period.

4.1.5 Correlation and Message Processing

After due dates for both regular declaration messages and manual correction messages have been passed, BAZG starts with correlating and processing of the accepted messages.

4.1.5.1 Correlation (to one or part of) a regular message

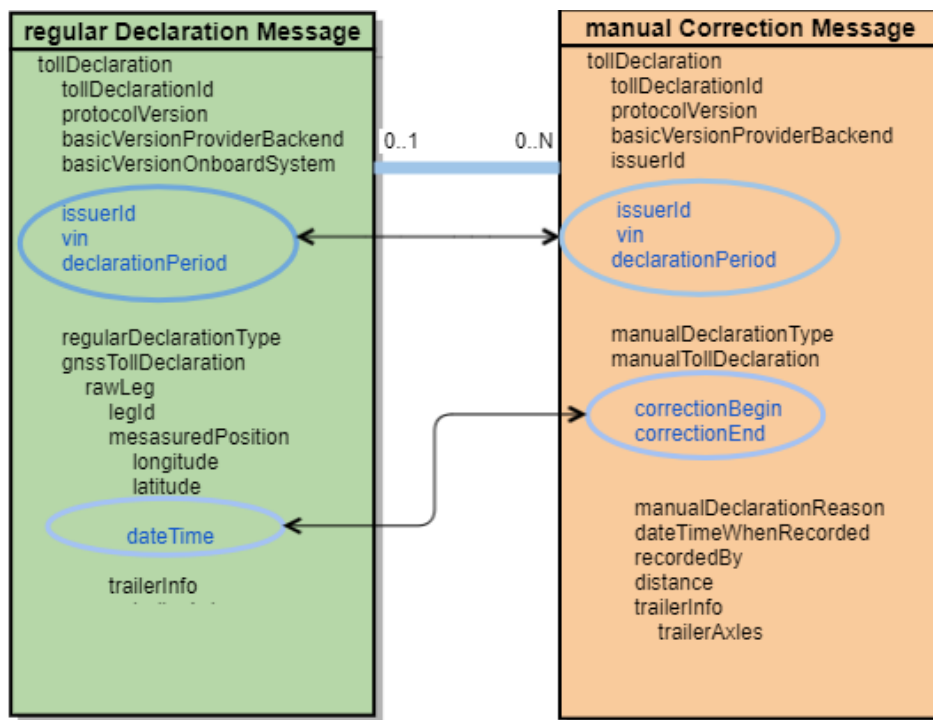
A manual correction message correlates to one (1) or part of a regular declaration message. Correlation is done via attributes:

- issuerId
- vin
- declarationPeriod

If referred to part of the regular declaration message, correlation is further limited by:

- correctionBegin
- correctionEnd

Note: A regular declaration message does not have to be existing (yet) when a corresponding manual correction arrives.



4.1.5.2 Correlation to regular messages and manuel corrections

- A correction of type correction vin is correlated to all affected reglar declaratons and manual corrections.

4.1.5.3 Correlation to exactly one regular message

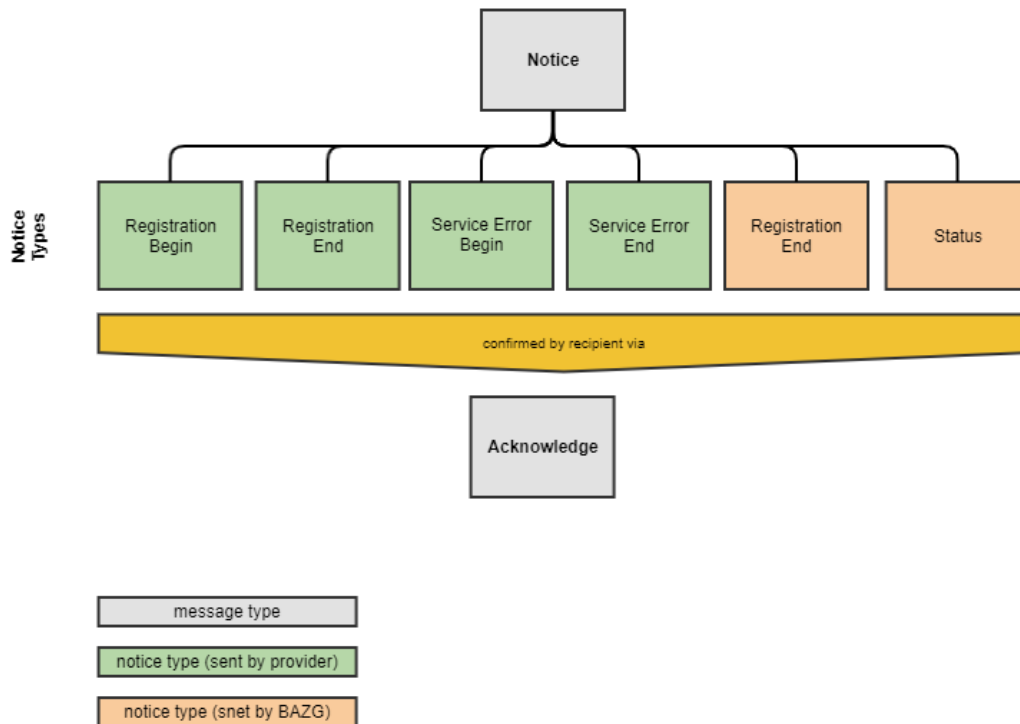
- In some cases, the specification requires correction to a single regular declaration.
- In this case a correlationId is used instead of correctionBegin / correctionEnd to identify the regular declaration to which the correction applies to.
- Manual correction message of type / reason SERVICE_REGISTRATION no correlation may exist.

4.1.5.4 Message Processing

- All received messages which have been sent to CAMIUNS until the 10th day following the relevant period are included in the processing.
- In cases, where multiple manual correction messages for a period have been received and accepted by CAMIUNS, all of them are taken into account.

4.2 Notice (Message + Interaction Pattern)

4.2.1 Notice Request



Notice Request Types

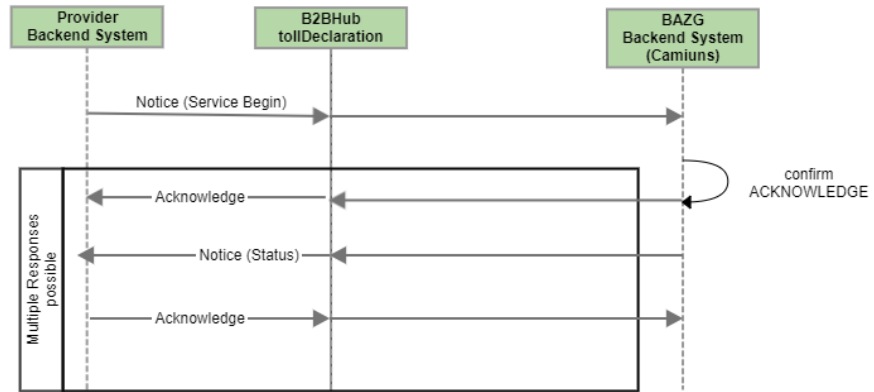
Request Type	Description
Registration Begin	Notice regarding an intended service begin for a vehicle.
Registration End	Notice of a registration end for the given vehicle.
Status	Response to an intended service begin. Provided for foreign vehicles only.

4.2.2 Acknowledge

The recipient of a notice needs to confirm each reception of a response with an acknowledge. The receipt of the response is hereby acknowledged.

Request Type	Description
Acknowledge	Confirmation or rejection of a received notice.

4.2.3 Interaction Pattern



A first response on business level is given usually within a few minutes to a hour. In cases of system downtime it might be delayed up to 24h. If an acknowledge is not provided within 24h, the message with must be resend with its original content after 24h / 48h. If failure still exists after 48h BAZG has to be contacted.

4.2.4 Correlation

Correlation of status messages can be done via vin.

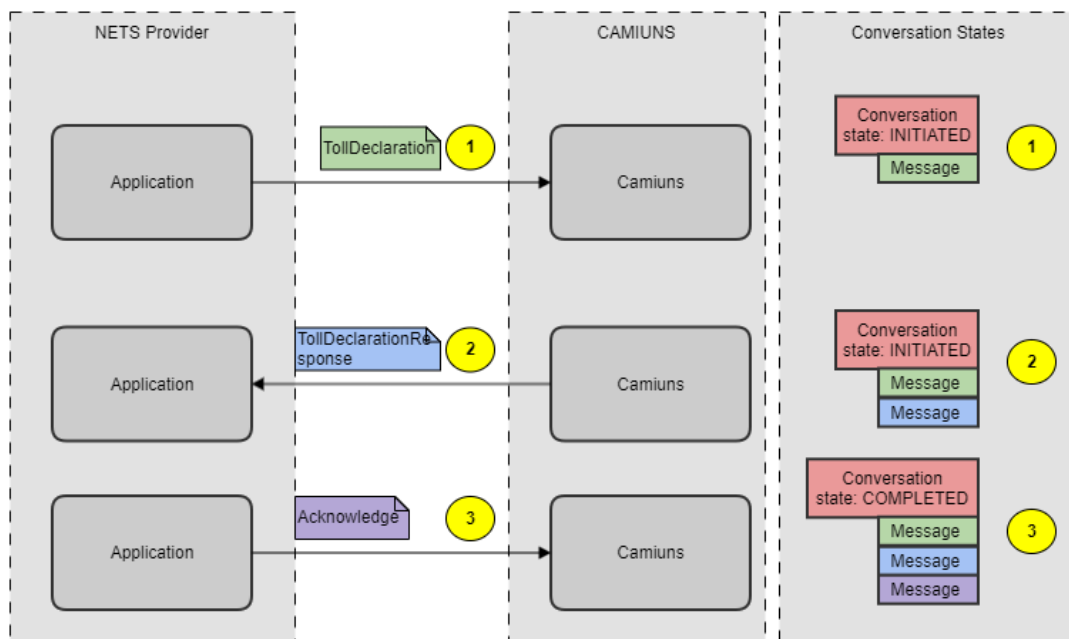
5 NETS Conversation Handling

Every message exchange triggers a new conversation. A conversation is completed when all messages have been adequately confirmed or answered.

If a conversation is open for too long, this is detected and reported via a corresponding channel. This obligation exists both on the provider's side and on the BAZG side.

5.1 TollDeclarations

5.1.1 Provider → BAZG

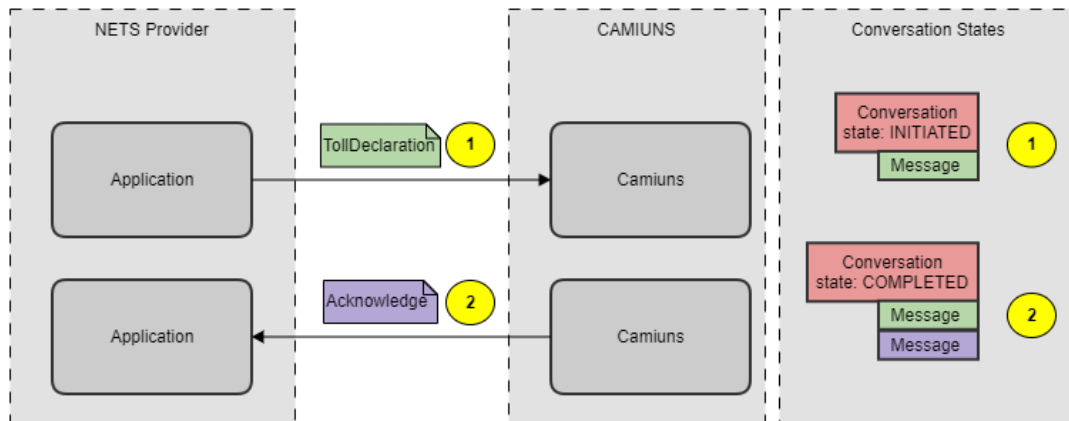


- 1** The NETS provider sends a TollDeclaration to BAZG. A new conversation is opened with the status INITIATED.
- 2** The TollDeclaration is checked and processed and is reported back to the NETS provider as accepted with a TollDeclarationResponse. The status of the conversation is not changed.
- 3** To prove that the provider has received the message, it sends an acknowledge message to the BAZG. The conversation is now COMPLETED.

Remarks:

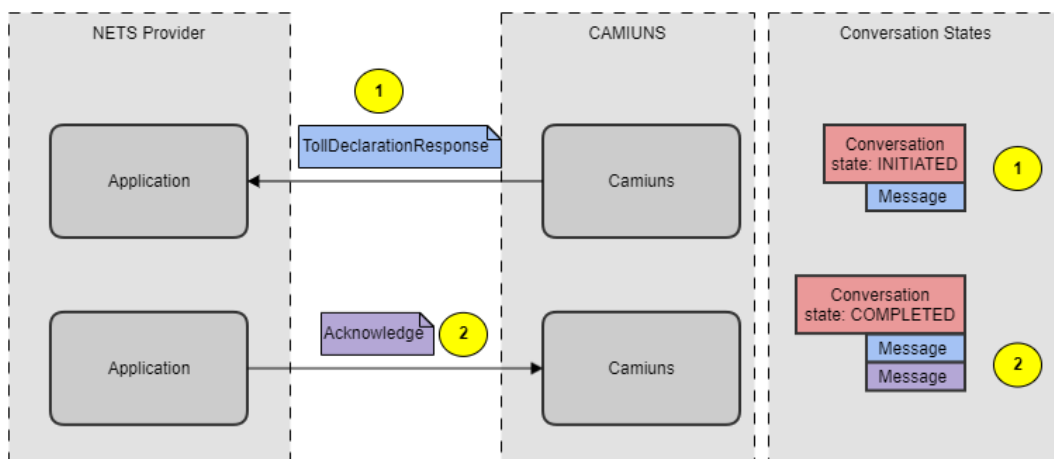
- if a manual toll declaration message is send (see for example [TCC01_01 CORRECTION VIN SUBMITTED INCORRECTLY](#)) based on a previous conversation, a new conversation is initiated. Every transmission of TollDeclaration messages leads to a new conversation.
- it doesn't matter if the acknowledge is negative, for the view of a conversation, the conversation is closed.

In case of some specific errors (schema validation, signature errors, the following scenario is also possible:



- 1 The NETS provider sends a TollDeclaration to BAZG. A new conversation is opened with the status INITIATED.
- 2 The structure of the xml message was invalid, so a negative acknowledge is send back. The conversation is now closed.

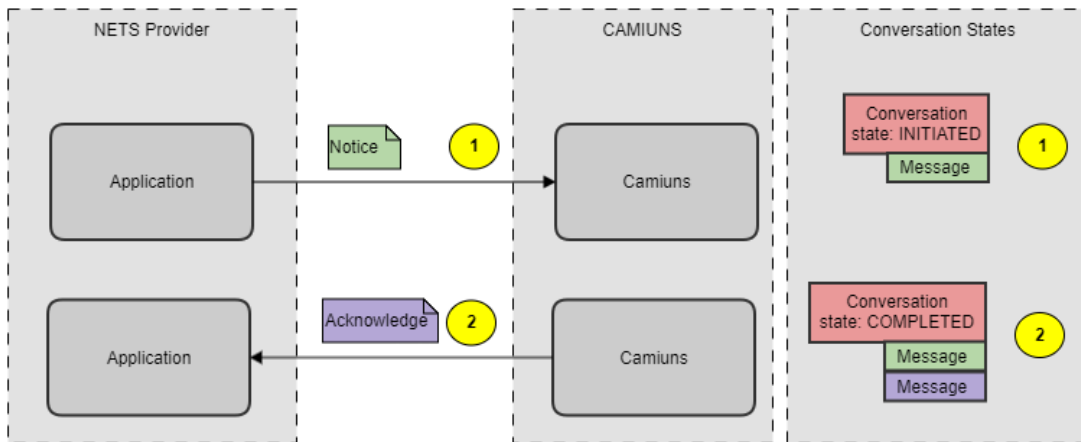
5.1.2 BAZG → Provider



- 1 A vehicle registration check is carried out approx. 3 days after receipt of a TollDeclaration. If the vehicle cannot be cancelled, for example, this is reported back to the NETS provider. A new conversation with INITIATED status is started.
- 2 To prove that the provider has received the message, it sends an acknowledge message to the BAZG. The conversation is now COMPLETED.

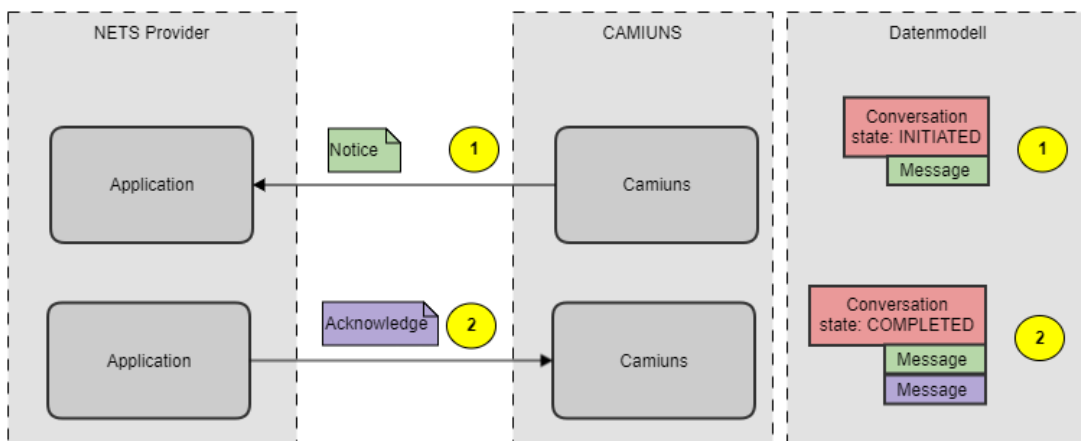
5.2 Notices

5.2.1 Provider → BAZG



- 1 The NETS provider has signed a contract with a new service user. He registers the service with a Notice to the BAZG. A new conversation is created in state INITIATED.
- 2 To prove that the BAZG has received the message, it sends an acknowledge message to the NETS provider. The conversation is now COMPLETED.

5.2.2 BAZG → Provider



- 1 (A provider wants to contract a NETS-A vehicle. To this end, he has sent a notice with REGISTRATION_BEGIN)
The BAZG has now checked whether the vehicle is registered in the Vehicle-Scs. The response is sent to the provider. For this reason, a new conversation is opened with the status INITIATED.

2 To prove that the provider has received the message, it sends an acknowledge message to the BAZG. The conversation is now COMPLETED.

6 NETS Messages

6.1 Message

Attribut	Multiplicity	L0	L1	L2	Description
message	1	X			Root-Element for a message
messageContent	1		X		Root-Element for the message content
contentHeader	1			X	Root-Element for the message content header
contentBody	1			X	Root-Element for the content body
ds:Signature	1		X		Root-Element for the enveloped signature

6.2 Message Content Header

Attribut	Multi- plicity	L0	L1	L2	Description
messageId	1	X			Unique identification number given by the issuer
messageDateTime	1	X			Date of the message creation given by the issuer
informationSenderId	1	X			Root-Element for sender information
issuerId	1		X		Businesspartner ID of the sender
informationRecipientId	1	X			Root-Element for recipient information
issuerId	1		X		Businesspartner ID of the recipient

6.3 Message Content Body

Attribut	Multi- plicity	L0	L1	L2	Description
notice	1	X			Identifier for the specific message type:
tollDeclaration					Notice: notice
acknowledge					Regular TollDeclaration: tollDeclaration Manual TollDeclaration: tollDeclaration Acknowledge: acknowledge

6.4 Notice

The table below shows a logical model.

Attribut	Multiplicity	L0	L1	L2	Description
notice	1	X			

Attribut	Multiplicität	L0	L1	L2	Description
GENERAL PART					
noticeId	1		X		Unique id issued by the issuer clearly identifying a specific tollDeclarationNotice. 0 to 2 ⁶³ -1 according to CEN/TS 16986:2016
noticeType	1		X		Type of notice: <ul style="list-style-type: none"> REGISTRATION_BEGIN REGISTRATION_END REGISTRATION_STATUS
dateTimeWhenRecorded	1		X		Contains the date and time when the notice has been recorded. Timeformat to be used is UTC (e.g. 2024-10-26T19:32:52Z).
recordedBy	1		X		The party who filed the declaration: <ul style="list-style-type: none"> HOLDER PROVIDER BAZG
REGISTRATION_BEGIN (PROVIDER → BAZG)					
registrationBegin	1		X		
vin	1		X		Vehicle identification number, vin according to the vehicle license.
tollDomain	1		X		If Vehicle is registered in CH or FL use CH else A. <ul style="list-style-type: none"> CH A
effectiveDate	1		X		Date when the service shall be / was activated.
REGISTRATION_END (BAZG → PROVIDER / PROVIDER → BAZG)					
registrationEnd	1		X		
vin	1		X		Vehicle identification number, vin according to the vehicle license.
effectiveDate	1		X		Last day of registration. Timeformat to be used is local date: 2024-01-29
reason	0..1		X		Possible Reasons for termination: <ul style="list-style-type: none"> PERMISSION_NO_LONGER_GRANTED DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS REGISTRATION_CANCELLED

Attribut	Multiplicit	L 0	L 1	L 2	Description
					<ul style="list-style-type: none"> SERVICE_NOT_TIMELY_ACTIVATED (in Diskussion)
REGISTRATION_STATUS (BAZG → PROVIDER)					
registrationStatus			X		
vin	1			X	Id correlating to the corresponding request.
isRegistered	1			X	true, false
effectiveDate	0..1			X	Date of latest registration / deregistration of vehicle.
comment	0..1			X	A comment regarding the Status

6.5 TollDeclaration (Regular Message)

The table below shows a logical model.

Attribut	Multiplicit	L 0	L 1	L 2	L 3	L 4	L 5	L 6	L 7	Semantics
tollDeclaration	1	X								Common root for the regular and manual interfaces.
tollDeclarationId	1		X							Unique id issued by the issuer clearly identifying a specific tollDeclaration. 0 to 2 ⁶³ -1 according to CEN/TS 16986:2016
protocolVersion	1		X							Current constant value: 0 Used to maintain different versions of this interface.
basicVersionProviderBackend	1		X							The basicVersionProviderBackend contains the software version of the backend used at the providers side to generate this request.
basicVersionOnboardSystem	0..1		X							The basicVersionOnboardSystem contains the software and hardware version of the onboardsystem used or to generate this request.
vin	1		X							Vehicle identification number, vin according to the vehicle license or primary source in the vehicle.

Attribut	Multiplicit	L 0	L 1	L 2	L 3	L 4	L 5	L 6	L 7	Semantics
declarationPeriod	1		X							Period, a single day to which the submitted data corresponds to.
regularTollDeclaration	1		X							
regularDeclarationType	1			X						GNSS = GNSS-data as recorded by the onboard system for the given period. WITHOUT_ACCESS = onboard system could not be accessed for the given period. VEHICLE_OUTSIDE_TOLL_DOMAIN = vehicle outside toll domain for the entire given period.
for tollDeclarations with regularDeclarationType = GNSS										
gnssTollDeclaration	0..1			X						Contains the entire position data as automatically recorded for the given period.
rawLegs	1				X					
rawLeg	1..N					X				Contains the entire position data for a leg. A leg represents a route passed with the same configuration of toll relevant parameters.
legId	1						X			Incremented integer value starting with 1.
measuredPositions	1						X			
measuredPosition	1..N							X		Position data shall be based on the WGS84 coordinate system. For details see: NIMA TR8350.2, Third Edition – Amendment 1, January 2000, Department of Defense – World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems, issued by the National Imaging and Mapping Agency (NIMA), US Department of Defense. Remark: There is no need to order positions in any given way.

Attribut	Multiplicity	L0	L1	L2	L3	L4	L5	L6	L7	Semantics
longitude	1								X	as defined in ISO 6709, in microdegrees, >0=east, <0=west, absolute value <=180°
latitude	1								X	as defined in ISO 6709, in microdegrees, >0=north, <0=south, absolute value <=90°
timestamp	1								X	date and time, shall be within reported period. Timeformat to be used is UTC.
trailer	0-1						X			Missing element = no trailer
trailerType	1							X		T=trailer, S=semitrailer
trailerWeight	1							X		0 = exempted from HVC Weight in kg with a maximum resolution of 10 kg. Note: In cases where the onboard system allows the user to enter a lower resolution than 10 kg, the weight shall be rounded downwards.

for tollDeclarations with regularDeclarationType = WITHOUT_ACCESS, no further attributes needed

for tollDeclarations with regularDeclarationType = VEHICLE_OUTSIDE_TOLL_DOMAIN, no further attributes needed

6.6 TollDeclaration (Manual Correction Message)

The table below shows a logical model.

Attribut	Multiplicity	L0	L1	L2	L3	L4	Description
tollDeclaration	1		X				
manual interface (partly automatically processed information)							
tollDeclarationId	1		X				Unique id issued by the issuer clearly identifying a specific tollDeclaration. 0 to 2 ⁶³ -1 according to CEN/TS 16986:2016
protocolVersion	1		X				Current constant value: 0

Attribut	Multiplicity	L0	L1	L2	L3	L4	Description
							Used to maintain different versions of this interface.
basicVersionProviderBackend	1		X				The basicVersionProviderBackend contains the software version of the backend used at the providers side to generate this request.
vin	1		X				Vehicle identification number, vin according to the vehicle license or primary source in the vehicle.
declarationPeriod	1		X				Period, single day to which the submitted data corresponds to.
manualTollDeclaration	1		X				Contains a correction for a related regular declaration or part of it.
manualDeclarationType	1			X			Type of the manual declaration: CORRECTION_TRACK CORRECTION_TRAILER CORRECTION_VIN CORRECTION_JOURNEY REVOCATION Details see chapter: Manual Declaraton Types & Reasons.
dateTimeWhenRecorded	1			X			Contains the date and time when the manual declaration has been filed by the recording party. Timestamp relevant for legal deadlines. Timeformat to be used is UTC (e.g. 2024-10-26T19:32:52Z).
recordedBy	1			X			The party who filed the declaration: <ul style="list-style-type: none"> HOLDER PROVIDER
manualDeclarationReason	1			X			Reason for the manual declaration as stated in chapter: Manual Declaration Reasons.
CORRECTION_TRACK							
correctionTrack	1			X			
correctionPeriod	1				X		
correctionBegin	1					X	dateTime UTC (e.g. 2024-10-26T19:32:52Z) / For further information about the usage of

Attribut	Multiplicity	L0	L1	L2	L3	L4	Description
							correctionBegin see remark at the end of this table.
correctionEnd	1					X	dateTime UTC (e.g. 2024-10-26T20:32:52Z) / For further information about the usage of correctionEnd see remark at the end of this table.
distance	1				X		distance in km.
trailerInfo	0..1				X		Missing element = no trailer
trailerType	1					X	T=trailer, S=semitrailer
trailerWeight	1					X	0 = exempted from HVC Weight in kg with a maximum resolution of 10 kg.
CORRECTION_TRAILER							
correctionTrailer	1			X			
correctionPeriod					X		
correctionBegin	1					X	dateTime UTC (e.g. 2024-10-26T19:32:52Z) / For further information about the usage of correctionBegin see remark at the end of this table.
correctionEnd	1					X	dateTime UTC (e.g. 2024-10-26T20:32:52Z) / For further information about the usage of correctionEnd see remark at the end of this table.
trailerInfo	0..1				X		Missing element = no trailer
trailerType	1					X	T=trailer, S=semitrailer
trailerWeight	1					X	0 = exempted from HVC Weight in kg with a maximum resolution of 10 kg.
CORRECTION_VIN							
correctionVin				X			
replacementVin	1				X		The vin used for correction.
CORRECTION_JOURNEY							
correctionJourney				X			
correctionBegin	1					X	UTC (e.g. 2024-10-26T19:32:52Z) / For further information about the

Attribut	Multiplicity	L0	L1	L2	L3	L4	Description
							usage of correctionBegin see remark at the end of this table.
correctionEnd	1				X		UTC (e.g. 2024-10-26T19:32:52Z)/ For further information about the usage of correctionEnd see remark at the end of this table.
REVOCACTION							
revocation				X			
tollDeclarationId	1				X		tollDeclarationId of manual correction which shall be revoked.

Remark: Usage of correctionBegin and correctionEnd.

Time within the provided declarationPeriod in the format hh:mm.ss. The combination of correctionBegin and correctionEnd can span over:

- a part of a day
- a whole day

The used resolution is:

- exact time where the begin and/or end can be exactly be determined.
- a quarter of an hour where time set is an approximation.

When the correction spans:

- the whole period, the correctionBegin has to be set to: 00:00:00 and for correctionEnd to 23:59:59,
- an approximate time span correctionBegin is including the starting second: e.g. 09:00:00 and exludes the end second: e.g. 09:14:59.

Declarations containing a CorrectionBegin and CorrectionEnd may not overlapp. If overlapping, the interface will respond with: validation_error, reason text: Timespan overlapping. The user shall be guided during data entry.

6.6.1 Manual Declaration Reasons

manualDeclaration Reason	Recorded By	Recommendation for the presentation of supporting information and validations within the failover solution	Notes
CORRECTION_TRACK			
SERVICE_ERROR	HOLDER	The system of the provider	The correction is due to a journey

manualDeclaration Reason	Recorded By	Recommendation for the presentation of supporting information and validations within the failover solution	Notes
		marks the timeframe marks confirmed errors by the begin of the error and the end of the error.	during a period where the provider signaled a service error. This error needs to be signaled whenever: <ul style="list-style-type: none"> • driving data cannot be recorded by the onboard system. • the error exists over a period longer than 5 minutes on a single day.
SERVICE_REGISTRATION	HOLDER	The system of the provider limits the usage of the service reason code: The reason service registration shall only be applied from the time of the initial service registration for a vehicle until the service has been taken into operation by the holder.	The correction is due to receive the registration for the service while commissioning has not yet been completed. This reason has not to be used in cases where provisioning is done for replacment reasons. Use service error instead.
JOURNEY_WITHOUT_DEVICE	HOLDER	The system of the provider limits the correction to timeframes where there has	The correction is due to a journey without device.

manualDeclaration Reason	Recorded By	Recommendation for the presentation of supporting information and validations within the failover solution	Notes
		been no journey with the given device.	
CORRECTION_TRAILER			
WRONG_TRAILER	HOLDER	The system of the provider limits the correction to journeys reported by a regular declaration of type gnss.	The correction is due to a journey with wrong trailer settings. Corrections of type WRONG_TRAILER have to be aligned with a reported regular declaration of type gnss.
CORRECTION_VIN			
INCORRECT_VIN	PROVIDER / HOLDER	The reason for the error is due to a discrepancy of the filed registration at the road traffic office and the vin provided. Therefore the error can have multiple sources and reasons.	The correction may be necessary in cases where a declaration has been accepted with error, reason code no_registration_for_vin and a reason text "vin not found". Once the correction has been successfully applied, the corresponding declarations will be confirmed with accepted / ok. The corrections shall be applied to all corresponding incorrect declarations already sent. Provider has to

manualDeclaration Reason	Recorded By	Recommendation for the presentation of supporting information and validations within the failover solution	Notes
			send a declaration for each affected period.
CORRECTION_JOURNEY			
NO_JOURNEY	HOLDER	The correction begin and end shall always span the whole day.	The correction has been proactively filed by the holder for a known period with no journey. This for periods where there are no journeys and the holder can not ensure access to the device (e.g. holidays).
MOVED_WITHOUT_JOURNEY_VEHICLE_LOADED	HOLDER	The system of the provider limits the correction to journeys reported by a regular declaration of type gnss.	Example: Vehicle loaded on truck, on train.
MOVED_WITHOUT_JOURNEY_NON_TAXABLE_JOURNEY	HOLDER	The system of the provider limits the correction to journeys reported by a regular declaration of type gnss.	The correction is due to the fact that the onboard system / unit has been moved without taxable journey.
REVOCACTION			
REVOKE_MANUAL_CORRECTION	HOLDER	The system of the provider limits the revocation to manual	Revoke a previously sent manual correction. If the specified time limit of 10

manualDeclaration Reason	Recorded By	Recommendation for the presentation of supporting information and validations within the failover solution	Notes
		correction where deadline for processing has not yet been reached.	days after the relevant declaration period has been exceeded BAZG will answer with refused and deadline missed.

6.7 TollDeclarationResponse

The table below shows a logical model.

Attribut	Multi- plicity	L0	L1	L2	Description
tollDeclarationResponse	1	X			
tollDeclarationId	1		X		TollDeclarationId to which this response correlates to.
responseSequenceNumber	1		X		Sequence number starting with 1 of the responses for the correlated tollDeclarationId. (Example: An initial response ACCEPTED can be overruled by a later response ACCEPTED_WITH_ERROR in case where a tollDeclaration from a different provider has been received).
responseType	1		X		ACCEPTED - Declaration has been accepted by BAZG. ACCEPTED_ERROR - Declaration has been accepted by BAZG with a signalled error. Action to be taken by issuer. REFUSED - Declaration has been refused by BAZG. Initial Message needs correction and potential resend of the corrected message.
responseReason	1		X		according to details in chapter "ResponseReasons".
responseReasonType	1			X	ResponseReasonID according to chapter "ResponseReasons".
responseReasonText	0..1			X	Further explanations for the ResponseReasonID.

6.7.1 ResponseReasons

ResponseType	ResponseReasonType	Description
ACCEPTED	OK	No action required. Note: Can be REFUSED due to new facts within 10 days.
ACCEPTED_ERROR / REFUSED	NO_REGISTRATION_FOR_VIN	No registration could be found for the provided vin. Response overrules a previously sent response with ResponseType ACCEPTED in cases where the confirmation of registration at the traffic office does not arrive within the expected time limit. 10 days after declaration: If the vin is not corrected in a timely manner, declarations filed under the incorrect vin will be REFUSED. For foreign vehicles no vehicle with the provided replacementVin has been registered for the NETS service.
REFUSED	DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS	BAZG has received declarations for the same vin from a different provider. Note: BAZG will accept the declarations for the day of provider / service change and refuse declarations for the following day for the provider where the vehicle has been under service previously.

ResponseType	ResponseReasonType	Description
REFUSED	DEADLINE_MISSED	Deadline for the declaration or correction that has been missed.
REFUSED	DECLARATION_ID_NOT_UNIQUE	<p>The combination of IssuerId and tollDeclarationId is not unique.</p> <p>The service provider is ordered to fix the error and potentially resend the message.</p>
REFUSED	REPLACEMENT_VIN_UNKNOWN	<p>No registration could be found for the provided replacement VIN.</p> <p>For domestic vehicles the replacementVin does not correspond to a registration filed by a traffic office or registration has been cancelled.</p> <p>For foreign vehicles no vehicle with the provided replacementVin has been registered for the NETS service.</p>
REFUSED	VALIDATION_ERROR	<p>Any general validation error. The error needs to be fixed by the provider and the necessary action has to be taken to resend the message.</p> <p>Note: Validation due to there nature cannot be handled on business level are supplied via the acknowledge message.</p>

6.7.2 ResponseReasonsText

The reason text in the message is explanatory in nature. In cases where the exact reason cannot be identified, the BAZG should be contacted.

6.8 Acknowledge

The table below shows a logical model.

Attribut	Multi- plicity	L0	L1	L2	L3	Description
acknowledge	1	X				
correlationId	1		X			The messageId of the original message this acknowledge correlates to.
ackCode	1		X			The acknowledge code. OK NOT_OK
issues	0..1		X			Root element of possible issues
issue	1..n			X		Root element of a single issue
issueCode	1				X	A possible issue code (has to be defined)
issueText	0..1				X	A optional text describing the issue.

6.8.1 Issue Codes

The table below contains possible issue codes in an acknowledge can occur.

- Issue Codes of type GENERAL can be set on both sides (from the provider or BAZG)

type	IssueCode	Description
GENERAL	1000	There are validation errors there are in contradiction to defined xsd schema.
GENERAL	1001	The signature is not valid.
GENERAL	1002	Issuer ID unknown
NOTICE	2000	The Notice cannot be accepted.

6.9 Schemas

The Schemas for the interfaces are under version control and are provided via [Test-Container.](#)

7 NETS Messages-Exchange Scenarios

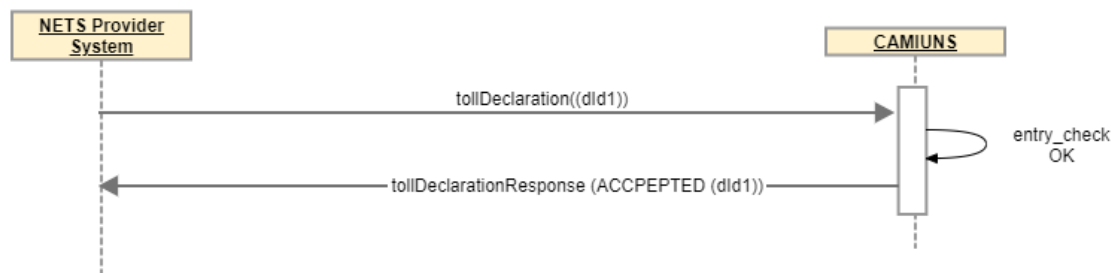
The following illustrations show some message exchange examples.

Note: Conversations are simplified in terms that the exchange of the necessary acknowledges for each message were omitted.

7.1 TollDeclaration (OK)

7.1.1 With result ACCEPTED

- The system of the NETS provider sends a declaration with tollDeclarationId (dId1).
- CAMIUNS entry check evaluates OK and returns ACCEPTED.



7.2 TollDeclaration (CORRECTION_JOURNEY)

There are 3 possible scenarios to correct a regular toll declaration with a manual correction of type correction journey:

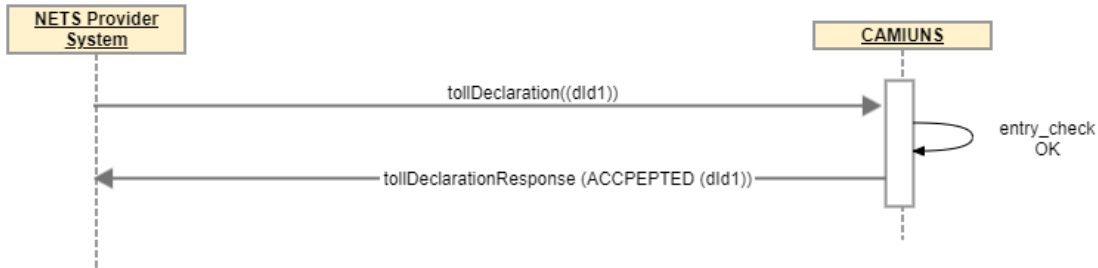
Type	Possible reason	Time period
NO_JOURNEY	The correction has been proactively filed by the holder for a known period with no journey. This for periods where there are no journeys and the holder can not ensure access to the device (e.g. holidays).	Day
MOVED_WITHOUT_JOURNEY_VEHICLE_LOADED	Example: Vehicle loaded on truck, on train.	Timespan
MOVED_WITHOUT_JOURNEY_NON_TAXABLE_JOURNEY	Possible reason: OBU is sent by mail from A to B and records a journey	Timespan

7.2.1 Preconditions

The system of the NETS provider sends a regular declaration as described in [TollDeclaration \(OK\)](#).

7.2.2 Sequence diagram

- The system of the NETS provider sends a manual declaration with tollDeclarationId (dId1).
- CAMIUNS entry check evaluates OK and returns ACCEPTED.



7.3 TollDeclaration (CORRECTION_TRACK)

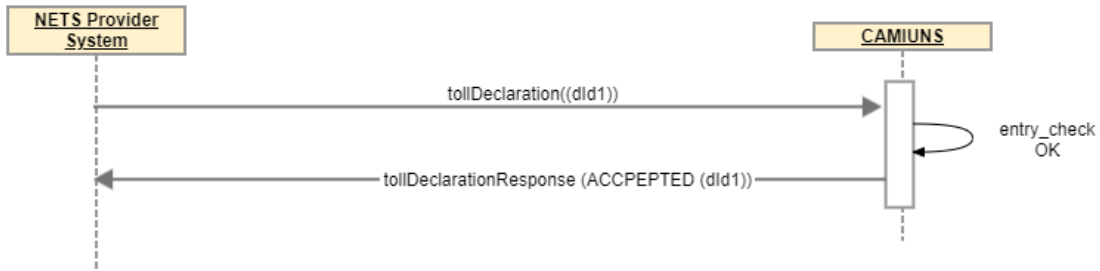
There are 3 possible scenarios to correct a regular toll declaration with a manual correction of type correction journey:

Type	Possible reason	Time Period
SERVICE_ERROR	<p>The correction is due to a journey during a period where the provider signaled a service error. This error needs to be signaled whenever:</p> <ul style="list-style-type: none"> • driving data cannot be recorded by the onboard system. • the error exists over a period longer than 5 minutes on a single day. 	Timespan
SERVICE_REGISTRATION	<p>The correction is due to receive registration for the service while commissioning has not yet been completed.</p> <p>This reason has not to be used in cases where provisioning is done for replacement reasons. Use service error instead.</p>	Timespan
JOURNEY_WITHOUT_DEVICE	The correction is due to a journey without device.	Timespan

7.3.1 Preconditions

The system of the NETS provider sends a regular declaration as described in [TollDeclaration \(OK\)](#).

7.3.2 Sequence diagram



7.4 TollDeclaration (CORRECTION_TRAILER)

The correction is due to a journey with wrong trailer settings. Corrections of type WRONG_TRAILER have to be aligned with a reported regular declaration of type gnss.

7.4.1 Preconditions

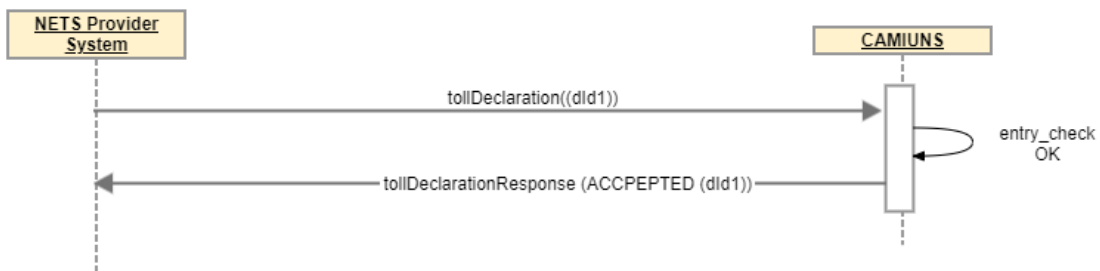
The system of the NETS provider sends a regular declaration of type gnss as described in [TollDeclaration \(OK\)](#).

dateTime of first gnss point	dateTime of last gnss point
2022-02-11T09:48:54.000Z	2022-02-11T12:59:59.999Z

7.4.2 Correction timespan

The timespan a correction can take place is limited by the underlying regular toll declaration (in the example above between 2022-02-11T09:48:54.000Z and 2022-02-11T12:59:59.999Z)

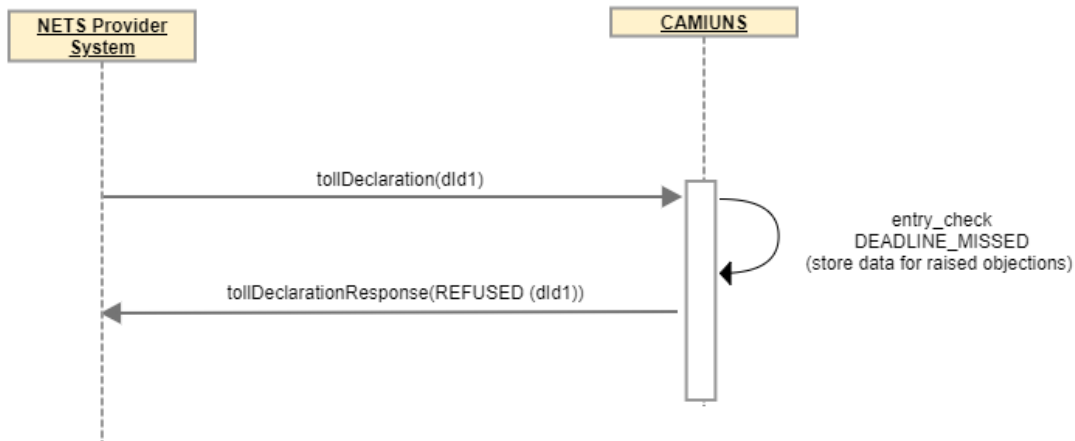
7.4.3 Sequence



7.5 TollDeclaration (DEADLINE_MISSED)

7.5.1 With result REFUSED

- The system of the NETS provider sends a declaration after the deadline given.
- During entry check CAMIUNS recognizes a DEADLINE_MISSED.
- CAMIUNS rejects the message with REFUSED and DEADLINE_MISSED.

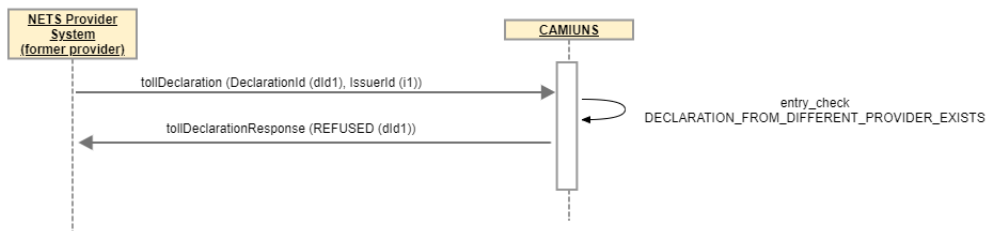


Note: even that regular declarations will be proceeded only for 10 days after declarationPeriod all regular declarations are expected to be sent to BAZG.

7.6 TollDeclaration (DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS)

7.6.1 With result REFUSED

- CAMIUNS has received a declaration for the same vehicle from a different, new provider. Therefore the messages of the existing old / former provider will be refused the day after.



Note:

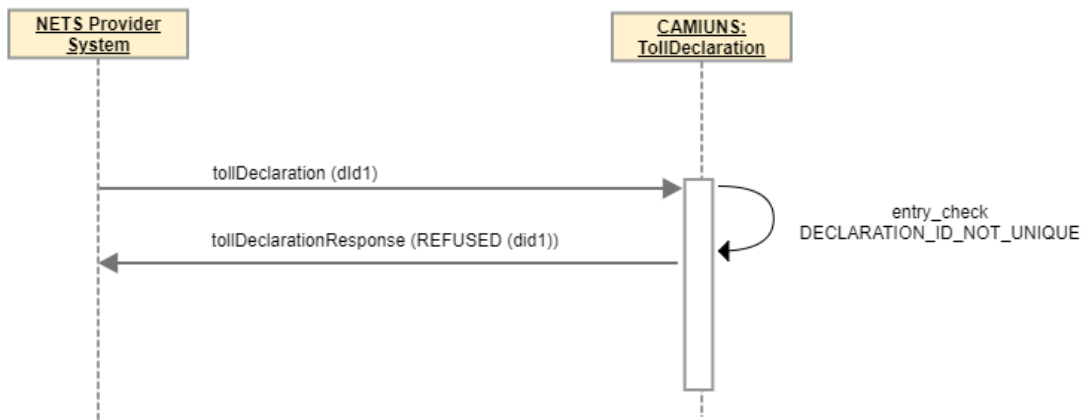
- Additionally a [notice REGISTRATION END](#) will be sent to the former provider in a different process, signaling that the delivery of further messages for the given vehicle must be stopped and remuneration will be terminated.
- The message of type DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS may be immediate or a previously accepted message may be refused.

7.7 TollDeclaration (DECLARATION_ID_NOT_UNIQUE)

7.7.1 With result REFUSED

- The system of the NETS provider sends a declaration with declaration id (did1).
- During entry check CAMIUNS recognizes a DECLARATION_ID_NOT_UNIQUE.

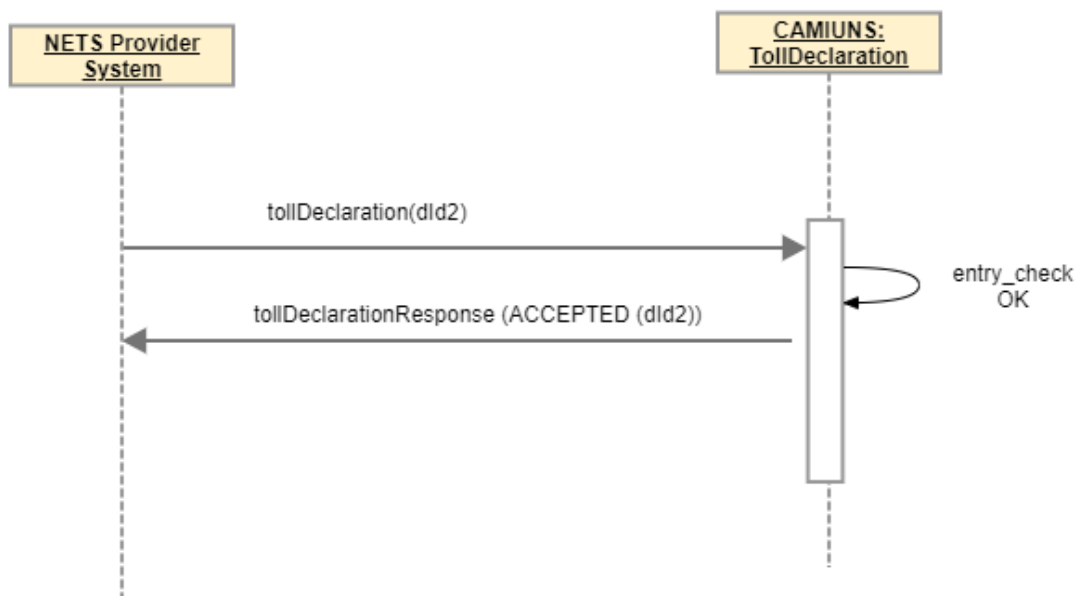
- CAMIUNS rejects the message with REFUSED and DECLARATION_ID_NOT_UNIQUE.



Note: Uniqueness is required within the declaration messages sent by one NETS provider.

7.7.1.1 resending with result ACCEPTED

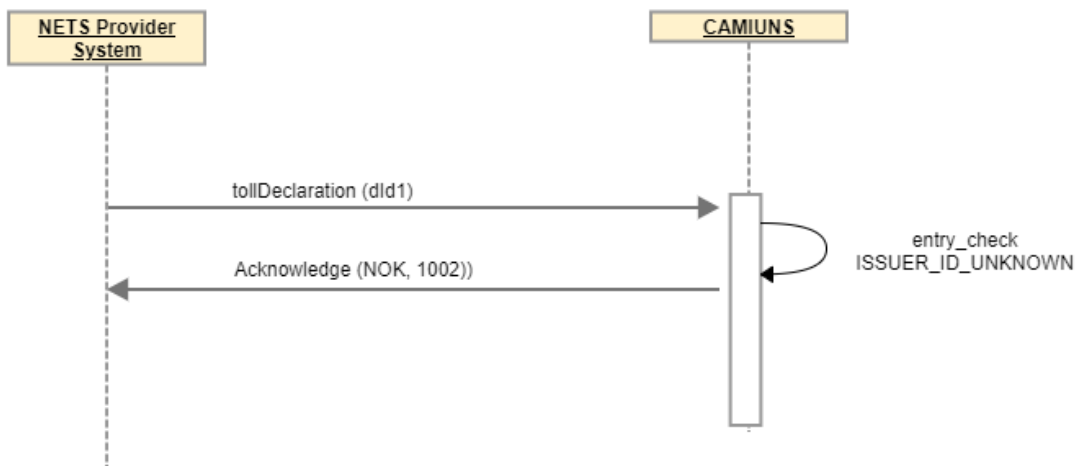
- provider successfully corrects DECLARATION_ID_NOT_UNIQUE



7.8 TollDeclaration, Notice (ISSUER_ID_UNKNOWN)

7.8.1 With result NOK (1002)

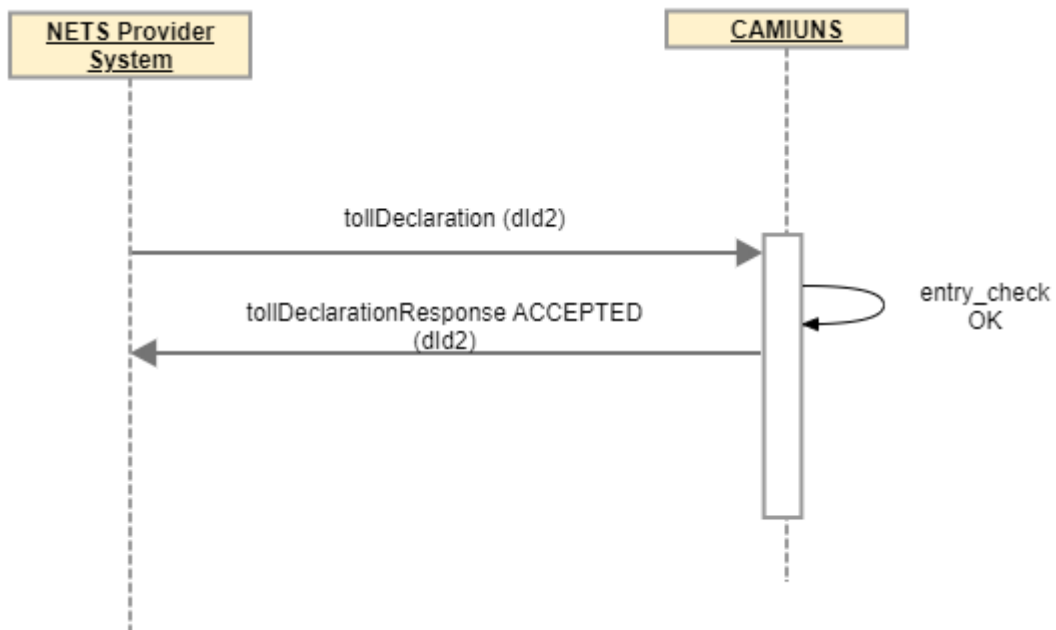
- The system of the NETS provider sends a declaration with declaration id (did1).
- During entry check CAMIUNS recognizes a ISSUER_ID_UNKNOWN.
- CAMIUNS rejects the message with REFUSED and ISSUER_ID_UNKONWN.



The scenario is also valid for messages of the type notice.

7.8.1.1 resending with result ACCEPTED

- The system of the NETS provider successfully corrects ISSUER_ID_UNKNOWN.
- Note: the same declarationId (did1) **cannot** be used to resend the message.

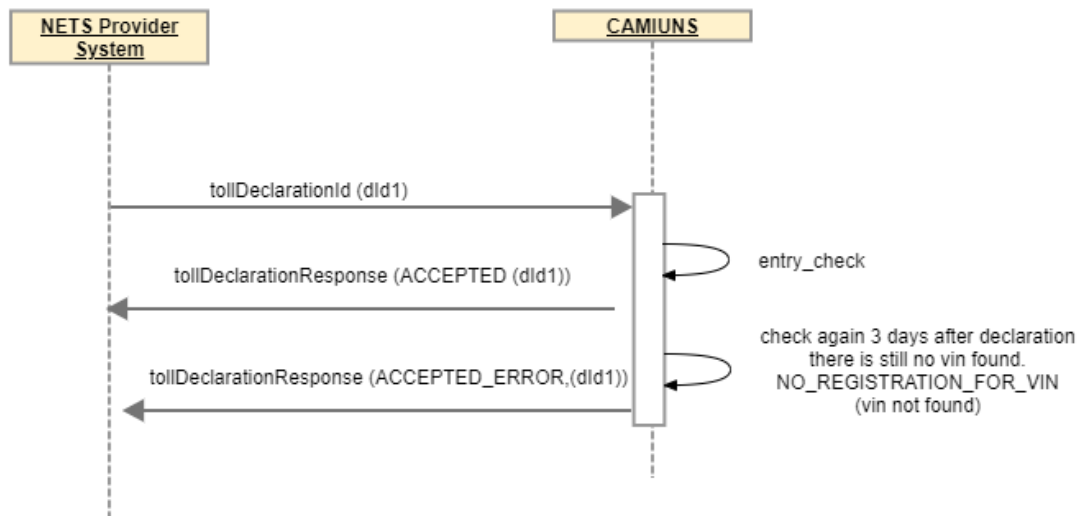


The scenario is also valid for messages of the type notice.

7.9 TollDeclaration (NO_REGISTRATION_FOR_VIN - vin not found)

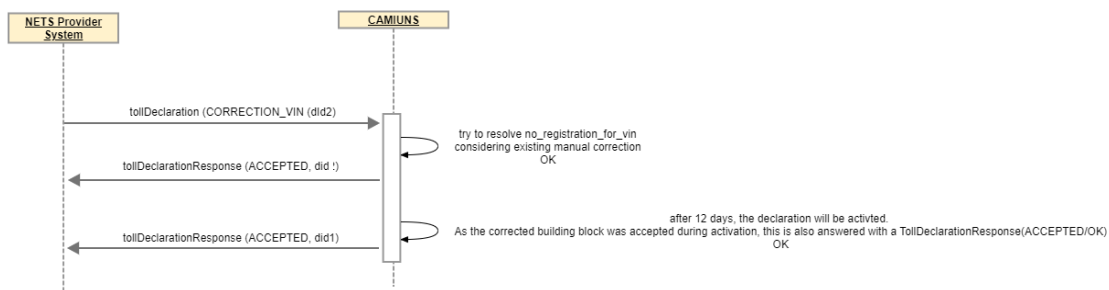
7.9.1 With result ACCEPTED ERROR / REFUSED

- During entry check CAMIUNS recognizes an NO_REGISTRATION_FOR_VIN.
- In expectation that this vin will be received successfully during the next coming days CAMIUNS confirms with ACCEPTED referring to (dId1).
- In the following days CAMIUNS tries to resolve the NO_REGISTRATION_FOR_VIN status.
- When not successful during 3-days, BAZG responds with an ACCEPTED_ERROR (NO_REGISTRATION_FOR_VIN) referring to (dId1). The reason text of the message will contain "vin not found"



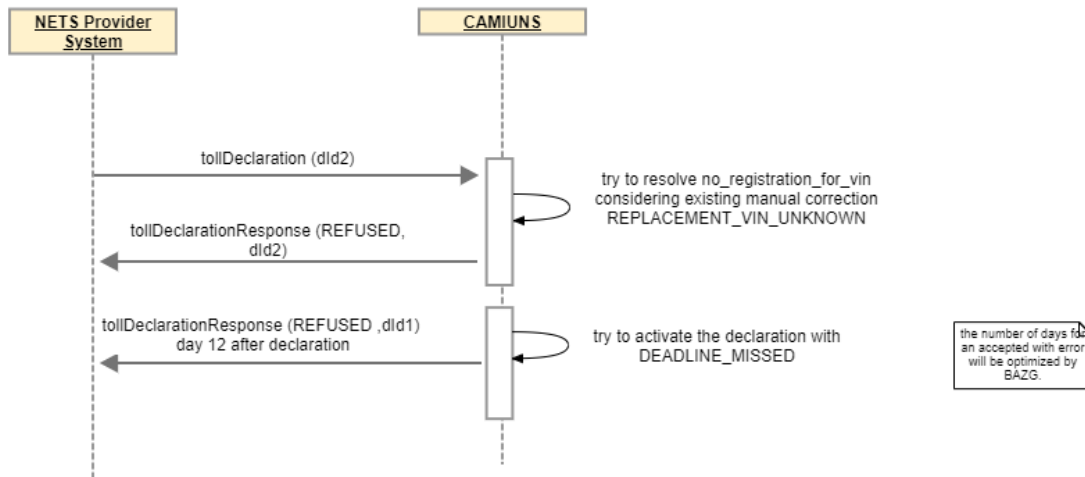
7.9.1.1 ... manual correction with result ACCEPTED

- The system of the NETS provider successfully corrects the NO_REGISTRATION_FOR_VIN.
- Both the manual correction (did2) and the referenced declaration (did1), see above, are confirmed with ACCEPTED.



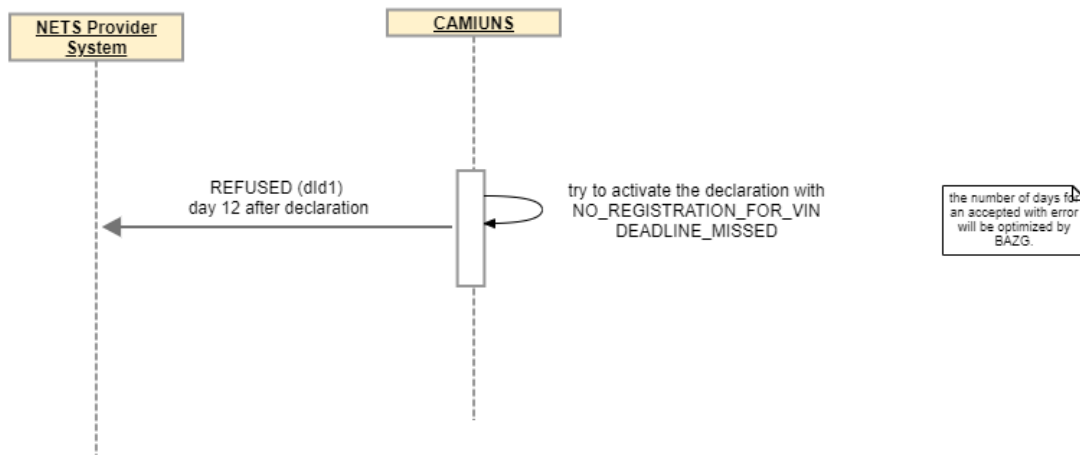
7.9.1.2 ... manual correction with result REFUSED

- The system of the NETS provider unsuccessfully corrects the NO_REGISTRATION_FOR_VIN.
- Both the manual correction (did2) and the referenced declaration (did1), see above, are REFUSED.



7.9.1.3 ... correction not timely received

- the provider / holder or traffic office misses to timely correct the ACCEPTED_ERROR (NO_REGISTRATION_FOR_VIN) status.
- the regular declaration (did) is REFUSED.

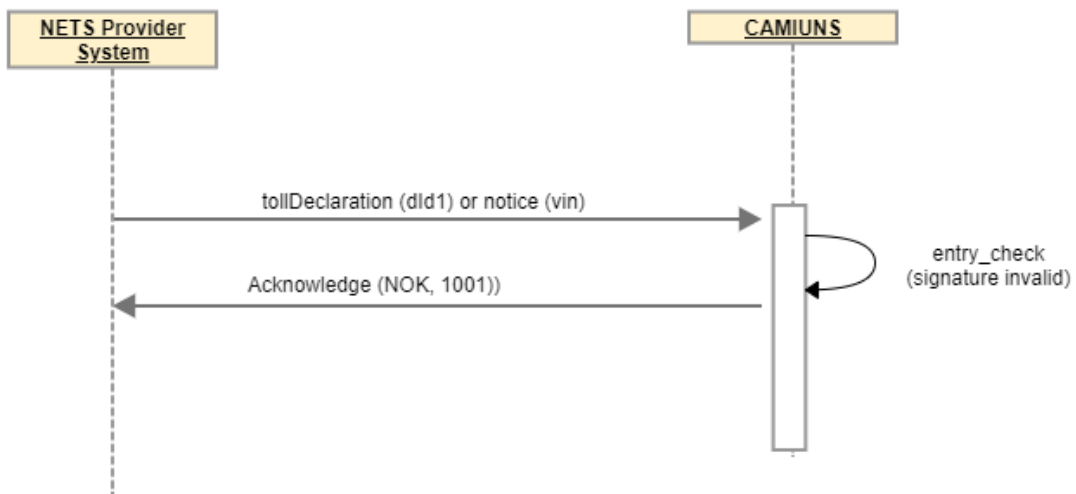


7.10 TollDeclaration, Notice (SIGNATURE_INVALID)

7.10.1 With result NOK (1001)

- The system of the NETS provider sends a regular declaration with declaration id (did1).

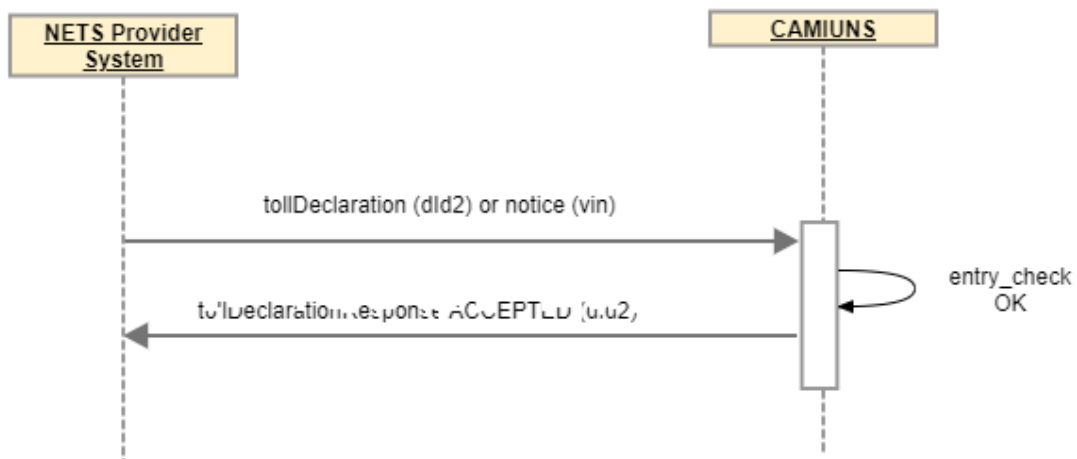
- During entry check CAMIUNS recognizes a ISSUER_ID_UNKNOWN.
- CAMIUNS rejects the message with REFUSED and ISSUER_ID_UNKONWN.



The scenario is also valid for messages of the type notice.

7.10.1.1 resending with result ACCEPTED

- The system of the NETS provider successfully corrects ISSUER_ID_UNKNOWN.
- Note: the same declarationId (did1) **cannot** be used to resend the message.



The scenario is also valid for messages of the type notice.

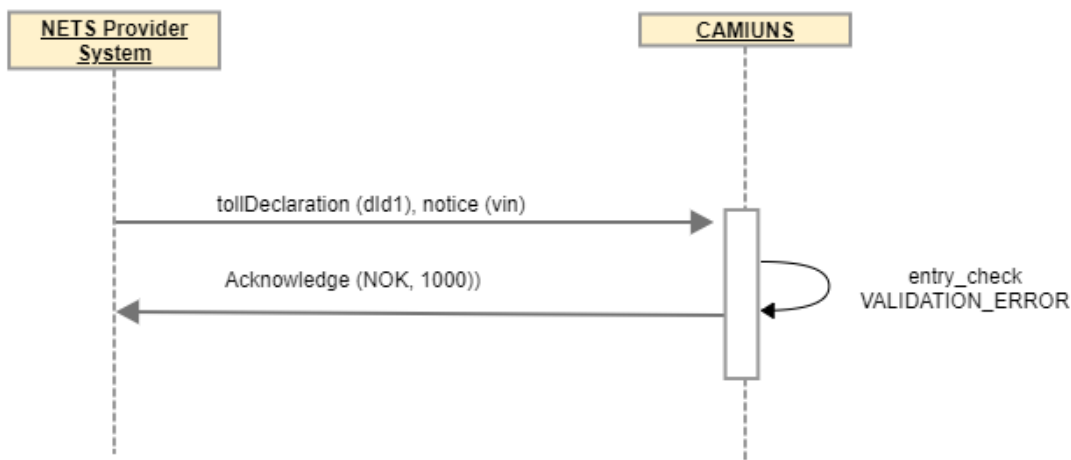
7.11 TollDeclaration, Notice (VALIDATION_ERROR)

Validation error can occur on two levels. The schema level and the application level. All errors which are on schema level, and therefore the corresponding message cannot be answered by BAZG with a correlating TollDeclarationResponse, are handled on schema level.

7.11.1 TollDeclaration, Notice (VALIDATION_ERROR Schema Level)

7.11.1.1 With result NOK (1000)

- The system of the NETS provider sends a regular declaration.
- An error occurs at schema level.
- CAMIUNS rejects the message with ACKNOWLEDGE NOK.

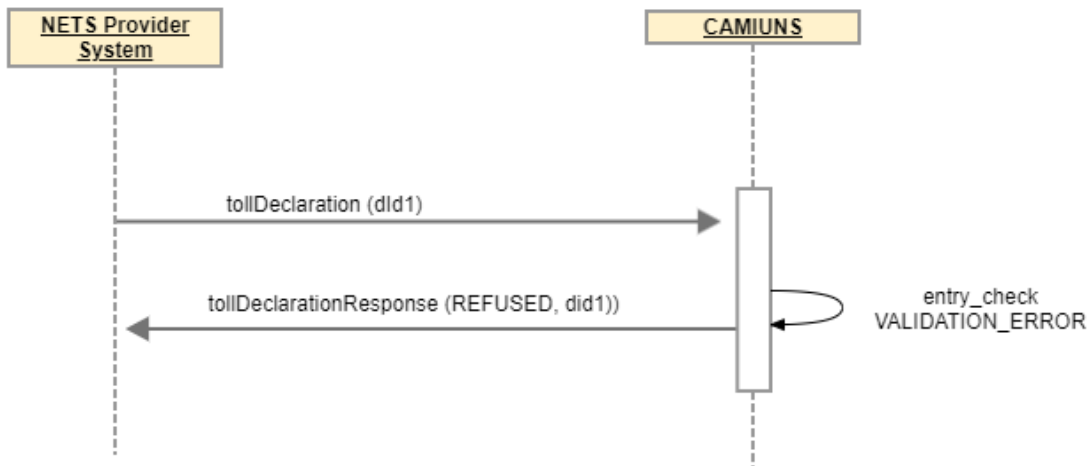


The scenario is also valid for messages of the type notice.

7.11.2 TollDeclaration (VALIDATION_ERROR Application Level)

7.11.2.1 With result REFUSED

- The system of the NETS provider sends a regular declaration.
- During entry check CAMIUNS recognizes a SCHEMA_VALIDATION_ERROR.
- CAMIUNS rejects the message with REFUSED and VALIDATION_ERROR. Validation error is used for different kind of general validation errors.
The field reason text holds the effective reason for the validation error.

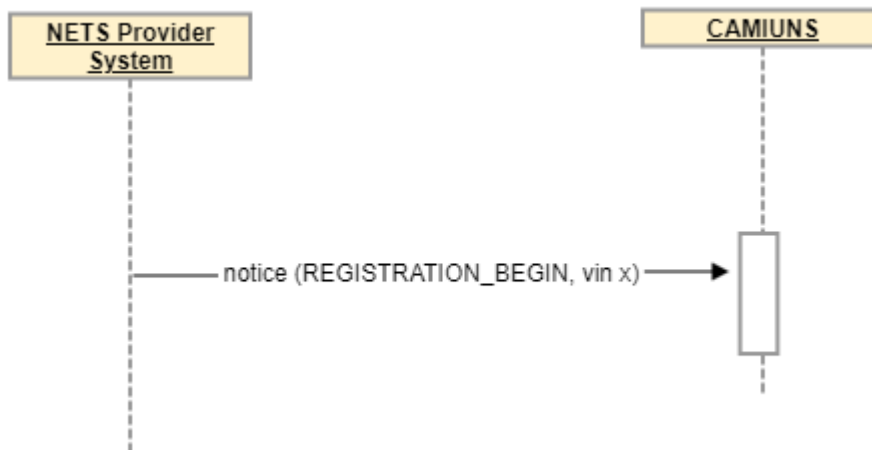


The scenario is also valid for messages of the type notice.

7.12 Notice (REGISTRATION)

7.12.1 Notice (REGISTRATION_BEGIN)

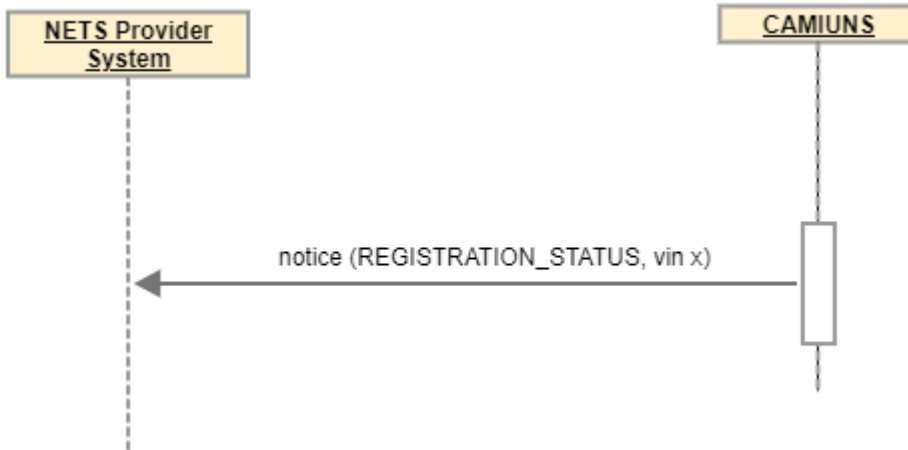
Once a provider wants to take a vehicle under service he issues a notice of type REGISTRATION_BEGIN for the corresponding VIN.



7.12.2 Notice (REGISTRATION_STATUS)

For notices REGISTRATION_BEGIN for vehicles of the tollDomain(CH) there will be no confirmation about the vehicle registration status. For notices REGISTRATION_BEGIN for vehicles of the tollDomain(A) CAMIUNS will provide a status of the requested vehicle.

Note: Vehicles for tollDomain(A) shall only be taken under service by the provider if confirmed by BAZG as registered and only after effective date.



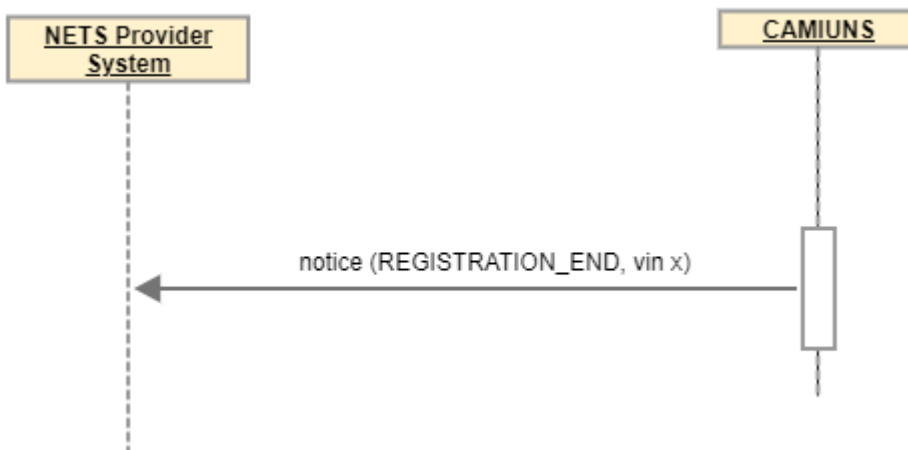
7.12.3 Notice (REGISTRATION_END)

7.12.3.1 Notice (REGISTRATION_END - Terminated by BAZG)

In cases where BAZG recognizes a reason to terminate the service for a given vehicle it sends a notice of type registration end and corresponding reason:

Following reasons are possible:

- PERMISSION_NO_LONGER_GRANTED
- DECLARATION_FROM_DIFFERENT_PROVIDER_EXISTS
- REGISTRATION_CANCELLED



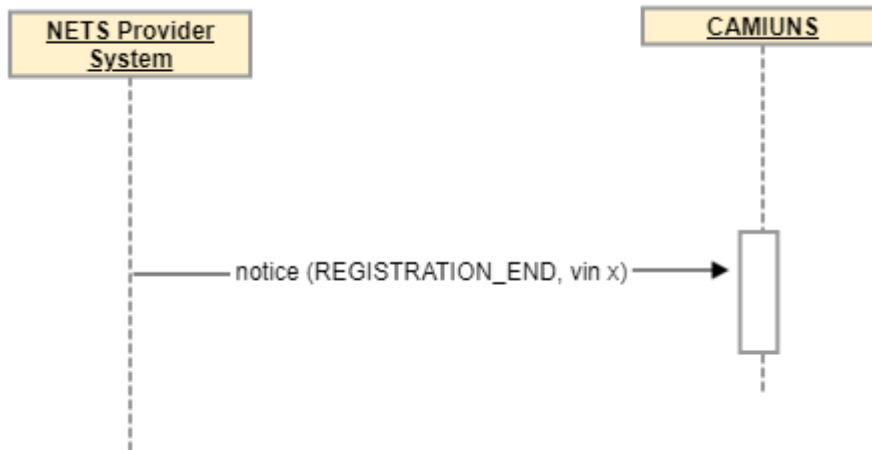
Note:

- Registration end is used by BAZG to signal the end of remuneration for a vehicle.

-
- A new REGISTRATION_BEGIN has to be sent, once a REGISTRATION for a vehicle has been signaled by REGISTRATION_END.

7.12.3.2 Notice (REGISTRATION_END - Terminated by the holder via NETS provider)

The service has been cancelled by the holder informing the provider. The provider therefore has to signal BAZG the termination of the service.



Note:

- Remuneration for a vehicle ends.
- A new REGISTRATION_BEGIN has to be sent if the service for the vehicle shall be continued.