

OWNER: DG TAXUD	ISSUE DATE: 06/03/2015	VERSION: 1.76-EN
TAXATION AND CUSTOMS UNION DG EMCS COMPUTERISATION PROJECT PHASE 3		
SUBJECT:		
ECP3-FITSDEV3-SC01-DDNEA		
DDNEA FOR EMCS PHASE 3		
(ECP3-FITSDEV3-SC01-DDNEA)		
FRAMEWORK CONTRACT TAXUD/2013/CC/121		
SPECIFIC CONTRACT NR 01		

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

DOCUMENT HISTORY

Edi.	Rev.	Date	Description	Action(*)	Sections
0	10	15/05/2008	First draft. Submitted for internal review.	I	All
1	00	21/05/2008	Incorporating internal review comments Submitted for review to DG Taxation and Customs Union.	I, R	As required
1	01	18/06/2008	Incorporating review comments Submitted for acceptance to DG Taxation and Customs Union.	I, R	As required
1	02	23/06/2008	Incorporating verification comments Re-submitted for acceptance to DG Taxation and Customs Union.	I, R	As required
1	03	25/06/2008	Implementation of ECP2- EMCSDEV-DDNEA-KEL-v1.07 Entry #7 and #8. Submitted for review to DG Taxation and Customs Union.	I, R	As required
1	04	02/07/2008	Incorporating review comments Submitted for acceptance to DG Taxation and Customs Union.	I, R	As required
1	05	04/07/2008	Transformed the Appendices from PDF format to Microsoft Word format. Submitted for information to DG Taxation and Customs Union	R	Appendices A, B, C, D, E, F and G.
1	06	08/07/2008	Updated Appendix B Codelists 5, 24, 47, 50, 60, 64 and 89. Submitted for review to DG Taxation and Customs Union.	R	Appendix B
1	07	29/07/2008	Incorporating review comments Submitted for acceptance to DG Taxation and Customs Union.	I, R	All Appendices
1	08	16/12/2008	Incorporating Workshop Decisions. Submitted for review to DG Taxation and Customs Union.	I, R	As required

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

1	09	21/01/2009	Incorporating review comments. Submitted for acceptance to DG Taxation and Customs Union.	I, R	As required
1	10	01/12/2009	Incorporating DDNEA RFCs #001, #002, #003, #004, #005, #006, #007, #008, #009, #010, #011, #012, #013, #014, #015, #016, #017, #020, #021, and #028. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	11	14/12/2009	Incorporating review comments. Submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	12	16/12/2009	Incorporating verification comments. Re-submitted for acceptance to DG Taxation and Customs Union.	R	Sections III.II.1.3.2 and III.IV.1
1	20	17/12/2009	Updated Section “Document Service Information” of the “Executive Summary”. Submitted for information to DG Taxation and Customs Union.	R	As required
1	21	11/03/2010	Incorporating DDNEA RFCs #019, #022, #023, #024, #025, #026, #027, #029, #030, #031, #032, #033, #034, #035, #036, #037, #038, #039, #040, #041, #042, #043, #044, #045, #047, #048, #049, #050, #052, #053, #054, #055, #056, #057, #058, #060, and #061. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	22	29/03/2010	Upon internal review, submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	30	31/03/2010	Submitted for information to DG Taxation and Customs Union.	I,R	As required

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

1	31	05/07/2010	Incorporating DDNEA RFCs #018, #046A, #046B, #051, #059, #062, #068, #069, #070, #071, #072, #073, #074, #075, #077, #078, #079, #080, and #081. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	32	20/07/2010	Upon internal review, submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	40	26/07/2010	Submitted for information to DG Taxation and Customs Union.	I,R	As required
1	41	29/12/2010	Incorporating DDNEA RFCs #082, #084, #088, #089, #090, #091, #092 and #095. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	42	14/01/2011	Upon internal review, submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	50	25/01/2011	Submitted for information to DG Taxation and Customs Union.	I,R	As required
1	51	04/02/2011	Removing implementation of DDNEA RFC #092. Re-submitted for information to DG Taxation and Customs Union.	I,R	As required
1	52	05/01/2012	Incorporating DDNEA RFCs #096, #100, #111, #112, #113, and #116. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	53	19/01/2012	Upon internal review, submitted for acceptance to DG Taxation and Customs Union.	-	-
1	60	30/01/2012	Submitted for information to DG Taxation and Customs Union.	-	-
1	61	28/03/2012	Incorporating DDNEA RFC #159. Submitted for information to DG Taxation and Customs Union.	I,R	Appendices H and I.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

1	62	15/06/2012	Incorporating DDNEA RFCs #063, #083, #085, #086, #087, #094, #097, #098, #099, #101, #102, #103, #104, #105, #106, #107, #108, #109, #110, #114, #115, #117, #118, #119, #120, #121, #124, #125, #126, #127, #128, #129, #131, #132, #133, #134, #135, #136, #137, #138, #139, #140, #142, #143, #144, #145, # 146, #147, #149, #150, #151, #152, #156, #157, #160, #161 and #162. Submitted for review to DG Taxation and Customs Union.	I,R	As required
1	63	09/07/2012	Incorporating review comments. Submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	64	11/07/2012	Incorporating verification comments. Re-submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	70	12/07/2012	Submitted for information to DG Taxation and Customs Union.	-	-

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

1	71	21/09/2012	<p>Incorporating internal review comments and MS verification comments:</p> <ul style="list-style-type: none"> • Removal of all references to Functional Stage 2 (or FS2); • Implementation of CP #6. Updates performed in order to clearly define the functionality in the case of the expiration of TIM_ACO and TIM_HIS timers; • Update of the title of section IV.IV.2; • Update of TC25: REQUESTED LIST OF CODES (FULL) in Appendix H (tcl.xsd) and in Appendix B to add the items ‘37 = Movement verification request reasons’ and ‘38 = Movement verification actions’; • Update of the simple types “MessageSenderType” and “MessageRecipientType” in the tms.xsd to also include “(CSMISE\EC)”. <p>Submitted for information to DG Taxation and Customs Union.</p>	I,R	As required
1	72	29/10/2013	<p>Incorporating DDNEA RFCs #163, #170, #172, #174 and #179.</p> <p>Submitted for review to DG Taxation and Customs Union.</p>	I,R	As required
1	73	13/11/2013	<p>Incorporating review comments.</p> <p>Submitted for acceptance to DG Taxation and Customs Union.</p>	I,R	As required
1	74	08/08/2014	<p>Incorporating DDNEA RFCs #066, #122, #153, #164, #166, #167, #168, #169, #173, #175, #176, #177, #178, #180, #181, #182, #183, #184, #185, #186, #187, #188, #189, #190 and #191.</p> <p>Submitted for review to DG Taxation and Customs Union.</p>	I,R	As required

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Document History	

1	75	19/09/2014	Incorporating review comments. Submitted for acceptance to DG Taxation and Customs Union.	I,R	As required
1	76	06/03/2015	Incorporating additional review comments. Submitted for information to DG Taxation and Customs Union.	I,R	As required

(*) Action: I = Insert R = Replace

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

EXECUTIVE SUMMARY

PURPOSE OF THIS DOCUMENT

This document is the **Design Document for National Excise Applications**. It specifies the design requirements to which any **Nationally Developed Excise Application (NDEA)** and **Centrally Developed Excise Application (CDEA)** needs to conform.

This applies to Nationally Developed Excise Applications (NDEA), developed by a Member State Administration (MSA) and to Centrally Developed Excise Applications (CDEA), developed by the Central Project Team (CPT).

This document is **applicable to every Nationally Developed Excise Application (NDEA)** and **Centrally Developed Excise Application (CDEA)** and must be considered as a mandatory document for all implementation and verification activities.

The purpose of this document is two-fold:

- To state unambiguously **what** needs to be developed. This will be achieved by specifying the sequences of Information Exchanges to be supported as a number of message exchange protocols and the detailed structure and building rules of these Information Exchanges.
- To define **how** the Information Exchanges need to be performed. Basically, every Information Exchange needs to be formatted (or represented) in XML representation and this formatted message needs to be transported between Excise Applications.

For NEAs, Information Exchanges are foreseen in the Common Domain (between Member State Administrations) and in the External Domain (between Member State Administration and Economic Operators).

This DDNEA mainly defines aspects for the *Application Level* and *Infrastructure Level* of EMCS Common Domain Architecture [A11]. In particular, the DDNEA provides all the required information in order for the NEAs to implement those architectural levels of the EMCS Common Domain Architecture. The *Business Level* is defined from FESS [A1] since it describes the expected “Services” for EMCS. However, the DDNEA considers this level and the FESS [A1] in order to define aspects of the *Application Level* since the Application Flow Control is defined based on the “Services” that are expected from EMCS for the Phase 3.

The EMCS Business Communication Channels that are in the scope of DDNEA are the following: BCC2, BCC7, BCC6, BCC9, BCC10, BCC11 and BCC12. The EMCS Infrastructure Communication Channels (ICC), which are in the scope of the DDNEA are the following: ICC4, ICC5, ICC6, ICC11, ICC15, ICC16 and ICC17.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

The DDNEA provides information how the NEAs shall communicate with the Common Domain Relay in order to use either the CCN/CSI Services or the CCN Intranet Services. In particular, the Section X - Transport of Messages via CCN/CSI specifies how the messages need to be transported across the CCN/CSI considering the BCC2, BCC7, BCC6, BCC10, BCC12 and their decomposition into ICC4, ICC5, ICC15 and ICC16. The Section XI - Transport of Messages via SOAP/HTTP defines the transport of messages via HTTP. These sections specify how a NEA can invoke the Common Domain Central Services Web Services or to exchange information via either a manual (web browser) or programmatic interface considering the BCC7, BCC6, BCC9, BCC10, BCC11 and BCC12 and their decomposition into ICC6, ICC11 and ICC17.

The DDNEA defines also the EMCS message format and structure. The Section IX-XML formatting describes how the EMCS messages need to be formatted in XML format. Moreover, some common design principles are provided regardless the transportation mechanism (CCN/CSI and SOAP/HTTP) in Section VIII-Design Principles. Finally, the EMCS Message Structure is defined in Section VII-Technical Message Structure. In this section, it is shown that DDNEA considers the EMCS Message Structure from both *Business Level* and *Application Level*. In particular, the Appendix D of DDNEA presents the Technical Message Structure including the FMS of FESS (Business Level) as well as some information required for the Application Flow Control in order to implement the Coordination Protocol.

Finally, the DDNEA specifies how the EMCS Common Domain Service Bus Interface defined in TESS [A11] can be implemented in order to achieve the *Business Process Choreography* and the *Application Flow Control*. Although, the *Business Process Orchestration* is mainly described in FESS [A1] through the Business Flow Diagrams and STDs, the DDNEA also includes some information for that concept of EMCS Common Domain Service Bus Interface. The concepts *Business Process Orchestration*, *Business Process Choreography* and the *Application Flow Control* are defined in TESS [A11].

Regarding *Business Process Orchestration*, the Section III-Core Business and Section V Central Services provide some information describing which operations should be executed in each domain for each specific scenario (Basic scenario, Change of Destination, etc.). However, the implementation of *Business Process Orchestration* is national issue since it refers to how the NDEA will handle the intra-domain exchanges. At least, the NDEA shall be aligned with the message exchange protocols and operations defined in the aforementioned sections.

The concepts *Business Process Choreography* and *Application Flow Control* are completely described in DDNEA through Section III-Core Business, Section V Central Services, Section VIII-Design Principles (VIII.I.3-Exception Handling) and Section VI- System Administration. The FESS defines the EMCS Business Processes. The controlling and coordination of these processes is described in the aforementioned sections.

The Section III and Sub-Section IV.IV clearly indicate what kind of validations (business or technical) shall be performed per case. In case of business validation, the Appendix D defines what Rules and Conditions shall be satisfied in order to ensure the business validity of the message, while for the case of technical validation, the Appendix H specifies the XSDs based on which the messages shall be checked in order to ensure their technical validity.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

Moreover, the DDNEA implements the *Coordination protocol* and the *Exception Handling of Application Flow Control* with the description of the message exchange protocols considering exceptional cases, the definition of state-machines per location (Dispatch or Destination), the definition of the error reporting depending on the type of error (format or functional) as well as some design constraints, and finally with the specification for the transport of messages via CCN/CSI (Section III, Sub-Section IV.IV, Section VIII and Section X). Finally, the logging activities that should be performed by the NDEAs are described in Section VI.

Apart from the technical aspects, the DDNEA also considers security aspects as these are defined in SESS [A10]. Therefore, the security is defined either at transport level or at message level depending on the transport mechanism (CCN/CSI, SOAP/HTTP, etc.). More information can be found for each transport mechanism in Section X, Section XI, and Section XII.

As a conclusion, the Header of the messages in Appendix D has been defined in order to contain all the required information for the *Coordination protocol*, the *Exception Handling* and the *Application Flow Control* in general (including security).

SCOPE OF THE DOCUMENT

DDNEA is restricted to the electronic Information Exchanges within EMCS.

This version of DDNEA is applicable to EMCS Phase 3. The functionality, which is in the scope of this Phase, is defined in the Scope of EMCS for Phase 3 [A2]. Finally, the DDNEA is aligned with the applicable documents listed in Table 1.

INTENDED AUDIENCE

The intended audience for this document includes:

- Any person responsible for the functional specifications of EMCS.
- Any person responsible for the development of software in the context of EMCS.
- Any person responsible for the definition of tests for EMCS.
- Anyone within the affected service suppliers in the CCN/CSI projects responsible for the delivery of the required services to EMCS.
- Any other authorised body concerned with EMCS, including the Excise Committee, the ECWP, the ECP steering committee, and professional organisations of economic Operators.

Audience is assumed to have a good understanding of the IT concepts and terminology used in this document. Also, it is assumed that audience is familiar with FESS [A1], the “Scope Document for EMCS” [A2], the TESS [A11] and the SESS [A10].

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

DOCUMENT STRUCTURE

This document is structured in sections and appendices. The sections are further subdivided in chapters. Selected sections are subdivided in sub-sections. The sections, sub-sections, chapters and appendices are summarised below:

SECTION I - GENERAL INFORMATION includes the following chapters:

- Chapter I.I.1 describes the **relationship of this document with other EMCS baseline documents**. It defines dependencies with these documents and states the applicability of these documents. It also specifies which standards have to be taken into consideration during the development and verification of any EMCS application.
- Chapter I.I.2 contains **definitions** used in this document (terminology, acronyms and abbreviations).
- Chapter I.I.3 describes the **symbolism and the conventions** used in the various models included in this document. It also discusses the technical naming conventions used for the data dictionary that has been included in this document.

SECTION II - SCOPE OF DEVELOPMENT discusses the items that need to be developed in EMCS Phase 3 applications. The respective Appendix A for EMCS Phase 3 accompanies this section.

The following sections contain a detailed definition of the **message protocols** to be supported for the different Business Processes in Excise. These message protocols are described by a collection of **Time Sequence Diagrams**, supported by **State Transition Diagrams**. Each section deals with one of the following Business Process areas:

SECTION III - CORE BUSINESS describes the Core Business of EMCS Phase 3. In particular, this section is subdivided in the following chapters:

- Sub-Section III.I defines the **Central Circuit Scenarios** and the possible message exchanges involved;
- Sub-Section III.II specifies the **Export Scenarios** and the possible message exchanges involved;
- Sub-Section III.III presents the **State Transition Diagrams** and specifically the valid states and state transitions;
- Sub-Section III.IV describes the **Functional Timers**;
- Sub-Section III.VI specifies the **Exception Handling** mechanism using some exceptional scenarios.

SECTION IV - FOLLOW-UP AND COLLABORATION describes the use case for administrative cooperation among MSAs. In particular, this section is subdivided in the following chapters:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

- Sub-Section IV.I defines the **Follow-up scenarios** for reporting events and controls that occur during excise movements;
- Sub-Section IV.II specifies the **Administrative cooperation scenarios** that describes the possible message exchange of spontaneous information and requests for assistance;
- Sub-Section IV.III presents the **Access to the history of movements** message exchange scenarios that may be used by the MSAs to access archived information for movements not available online;
- IV.II.3, IV.II.3, IV.III.2 and IV.IV.5 describe the **State Transitions** occurring during the execution of the *Administrative cooperation*, *Access to the history of movements* and *Movement verification for duty-paid movements* scenarios.

SECTION V - Central Services deals with the message exchange protocols between the NDEAs and the Common Domain Central Services applications, namely the System for Exchange of Excise Data (SEED) and the Central Services/Management Information System for Excise (CS/MISE). This section is subdivided as following:

- Sub-Section V.I - **SEED** describes the System for Exchange of Excise Data providing management and dissemination services of information on the Economic Operators and the Reference Data;
- Sub-Section V.II - **CS/MISE** describes the Central Services/Management Information System for Excise providing the monitoring and the reporting services on the operations of EMCS, which includes collecting, distributing and publishing EMCS business and technical statistics, by collecting and publishing information on movements (ARC follow-up), and monitoring the availability of all MSA applications (Availability Monitoring).

SECTION VI - SYSTEM ADMINISTRATION deals with issues such as logging and tracing and any other administration function to be foreseen.

SECTION VII - TECHNICAL MESSAGE STRUCTURE defines the detailed technical structure of the Information Exchanges of EMCS. For technical reasons, the technical message format is sometimes different from the logical format defined in FESS [A1]. This section is further subdivided as follows:

- Chapter VII.I.1 introduces the content of the Section VII.
- Chapter VII.I.2 introduces the **data dictionary**. It defines a number of items that make up a message, such as Data Items, Data Groups, and Codelists (sets of discrete values). This chapter is accompanied by Appendix G, Appendix F and Appendix A.
- Chapter VII.I.3 presents the detailed **Technical Message Structure** (TMS) for the different Information Exchanges. The detailed TMS for all messages is included in Appendix D. This chapter only explains how the Appendix D needs to be interpreted and used.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

- Chapter VII.I.4 describes the “Message Header” used in the Information Exchanges.
- Chapter VII.I.5 discusses the issue of **consistency**. It defines with which Excise documents this DDNEA needs to be consistent (such as “FESS” [A1]) and it explains how this consistency has been achieved during the TMS definition.

SECTION VIII - DESIGN PRINCIPLES explains how the system, defined in the previous sections, needs to be built. Basically, every Information Exchange needs to be formatted in XML and needs to be transmitted across one of the communications protocols (CCN/CSI and SOAP/HTTP). This section states a number of principles that are common regardless the transportation mechanism:

- Chapter VIII.I.1 discusses the overall **approach**.
- Chapter VIII.I.2 discusses the usage of **character sets** and **Data Item conventions**.
- Chapter VIII.I.3 defines technical details for the **exception handling** in EMCS Phase 3.
- Chapter VIII.I.4 defines **constraints** (any restrictions that are applicable to EMCS development).

SECTION IX - XML FORMATTING defines how messages need to be formatted in XML format. This section is structured as follows:

- Chapter IX.I.1 defines **XML conventions** for EMCS.
- Chapter IX.I.2 specifies the **Character Sets** that shall be supported by the NDEAs.
- Chapter IX.I.3 discusses the XML formatting of the **Information Exchanges**. Appendix E accompanies this chapter.
- Chapter IX.I.4 discusses the **XML Schemas** of EMCS messages. Appendix H accompanies this chapter.

SECTION X - TRANSPORT OF MESSAGES VIA CCN/CSI defines how messages need to be transported across the CCN/CSI communication platform. This section is subdivided as follows:

- Chapter X.I.1 defines **architectural assumptions** made for the transport of messages via CCN/CSI and details where references to CCN/CSI can be found.
- Chapter X.I.2 presents the **mandatory CCN/CSI elements** that will ensure end-to-end communication between two CCN gateways.
- Chapter X.I.3 presents the **recommended CCN/CSI elements** for sending and receiving messages.
- Chapter X.I.4 defines the **configuration information** necessary for the CCN gateways.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

SECTION XI - TRANSPORT OF MESSAGES VIA SOAP/HTTP defines the transport of messages via SOAP/HTTP that can be used by a NDEA to invoke the Common Domain Central Services Web Services. This section is divided as follows:

- Sub-Section XI.I - **Topology** describes the **topology** of the HTTP transport over the CCN, specifies the **URIs** for the Common Domain Central Services Web Services and defines the syntax for the **Web Service relative path**.
- Sub-Section XI.II - **CCN Configuration** describes the **CCN/HTTP configuration** in order for the Central Services Web Services to be accessible.
- Sub-Section XI.III - **Web Service Standards** provides information for the following **Web Service Standards**: SOAP, HTTP, XML-RPC, WSDL and WS-Security.
- Sub-Section XI.IV - **Recommended Usage** explains the requirements for a NDEA that communicates with the web services over the **SOAP/HTTP**.
- Sub-Section XI.V - **Web Service Transactions** provides information and describes the implication of **Web Services Transactions**.
- Sub-Section XI.VI - **Web Service Security** defines the **Web Services Security** at Transport and Message level.
- Sub-Section XI.VII - **CCN user HTTP authentication** specifies the **authentication** mechanism in the CCN/HTTP.

SECTION XII - APPLICATION AUTHENTICATION INTRANET SERVICES describes the services which provide HTTP interfaces so that the web applications can **login** (ccnServerLogin) and **logout** (ccnServerLogout) from CCN.

APPENDICES

- Appendix A presents all messages included in the scope of DDNEA for EMCS Phase 3.
- Appendix B contains a definition of all Codelists used for EMCS that are applicable for Phase 3.
- Appendix C presents how the different Data Groups and Data Items are correlated to the messages.
- Appendix D contains the definition of all messages for EMCS Phase 3.
- Appendix E contains the XML mapping of all Data Items and Data Groups of the EMCS messages.
- Appendix F and Appendix G contain a data dictionary for all elements (Data Items and Data Groups) used to construct these messages.
- Appendix H provides the XML schemas for all messages used in EMCS.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Executive Summary	

- Appendix I defines the WSDL files for the web services provided by the Central Services.
- Appendix J contains the catalogue of the Business Rules (BRs) which were identified in the Elementary Business Processes of FESS Section II: CORE BUSINESS.
- Appendix K contains, in an alphabetical order, a complete list of the applicable Rules and Conditions and the IE paths associated with them.

DOCUMENT SERVICE INFORMATION

The different parts that make up DDNEA will be submitted individually to configuration and version control. Individual components may be upgraded and delivered separately.

Maintenance will be provided for this document. The Taxation and Customs Union DG will define and schedule the different deliveries.

Comments can be submitted to this document, either via organised reviews or via calls to the EMCS Central Help Desk.

Known errors to this DDNEA will be maintained in the format of the Known Error List (KEL) published on the EMCS Central Help Desk.

Whenever a part of this document is referred to, a reference will be given either to an entire section or an entire chapter (within a section) or a paragraph (for any other subdivision).

This document will be submitted as a Word file with the following naming convention:

- **ECP3- FITSDEV3-SC01-DDNEA_P3-Vy.zz-EN.doc**, where y and zz are version and revision numbers.

All appendices of EMCS Phase 3 will be delivered as:

- **ECP3- FITSDEV3-SC01-DDNEA_P3_App_X_yzz-EN.DDD**, where X stands for the Appendix name, y and zz are version and revision numbers, and DDD is the document type (PDF for an Adobe Acrobat 5-file, DOC for MS Word, MDB for MS Access).

Please note that the appendices do not evolve separately from the main document. This means that the appendices will have the same version as DDNEA (**yzz**).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

RESUME

OBJECTIF DE CE DOCUMENT

Le présent document constitue la **documentation de conception pour les applications nationales d'accises** (Design Document for National Excise Applications ou DDNEA). Il définit les règles de conception auxquelles toute application de contrôle des accises doit se conformer.

Ceci concerne les applications d'informatisation des accises développées au niveau national (Nationally Developed Excise Applications ou NDEA) par l'administration d'un État membre (Member State Administration ou MSA) et les applications d'informatisation des accises développées au niveau central (Centrally Developed Excise Applications ou CDEA) par l'équipe centrale chargée du projet (Central Project Team ou CPT).

Ce document **s'applique à toutes les NDEA ainsi qu'à toutes les CDEA** et doit être considéré comme obligatoire pour toutes les activités de mise en oeuvre et de vérification.

L'objectif de ce document est double:

- Indiquer clairement quels sont les **éléments** à développer en définissant d'une part les séquences d'échanges d'information à prendre en charge comme un certain nombre de protocoles d'échange de message et d'autre part la structure détaillée et les règles d'organisation de ces échanges d'information.
- Définir **la manière** dont les échanges d'information doivent s'effectuer. Chaque échange d'information doit essentiellement être formaté (ou représenté) en une représentation XML puis ce message formaté doit être transporté d'une application d'accises à l'autre.

Pour les NEA, des échanges d'information sont prévus dans le domaine commun (entre administrations des États membres) et dans le domaine externe (entre administration d'État membre et opérateurs économiques).

Ce DDNEA définit en particulier différents aspects du *niveau application* et du *niveau infrastructure* de l'architecture du domaine commun EMCS [A11]. Le DDNEA fournit notamment toutes les informations nécessaires afin que les NEA puissent mettre en place ces deux niveaux de l'architecture du domaine commun EMCS. Le *niveau métier* est, quant à lui, déterminé à partir de la FESS [A1] car il décrit les «services» attendus d'EMCS. Le DDNEA prend cependant en compte ce niveau-ci et la FESS [A1] afin de définir différents aspects du *niveau application* car le contrôle des flux de l'application est déterminé en fonction des « services » attendus d'EMCS en phase 3.

Les canaux de communication EMCS de type métier (Business Communication Channels ou BCC) entrant dans le champ d'application du DDNEA sont les suivants: BCC2, BCC7, BCC6, BCC9, BCC10, BCC11 et BCC12. Les canaux de communication EMCS de type infrastructure (Infrastructure Communication Channels ou ICC) entrant dans le champ d'application du DDNEA sont les suivants ICC4, ICC5, ICC6, ICC11, ICC15, ICC16 et ICC17.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

Le DDNEA contient des informations sur la manière dont les NEA doivent communiquer avec le relais du domaine commun dans le but d'utiliser soit les services CCN/CSI soit les services Intranet CCN. La Section X-Transport de messages via CCN/CSI précise notamment comment les messages doivent être transportés à travers le réseau CCN/CSI en fonction des canaux BCC2, BCC7, BCC6, BCC10, BCC12 et de leur décomposition en canaux ICC4, ICC5, ICC15 et ICC16. La Section XI-Transport de messages via SOAP/HTTP définissent le transport de messages via HTTP. Ces sections spécifient comment une NEA peut faire exécuter les services web des services centraux du domaine commun ou échanger des informations par le biais d'une interface manuelle (navigateur web) ou programmatique en fonction des canaux BCC7, BCC6, BCC9, BCC10, BCC11 et BCC12 et de leur décomposition en canaux ICC6, ICC11 et ICC17.

Le DDNEA définit également le format et la structure du message EMCS. La Section IX-Formatage XML décrit la façon dont les messages EMCS doivent être convertis au format XML. Certains principes de conception communs à tous les mécanismes de transport (CCN/CSO ou SOAP/HTTP) sont par ailleurs exposés dans la Section VIII-Principes de conception. Enfin, la structure du message EMCS est définie dans la Section VII-Structure technique du message. Cette Section indique que, dans le DDNEA, la structure du message EMCS est envisagée à la fois au *niveau métier* et au *niveau application*. L'appendice D du DDNEA présente plus particulièrement la structure technique du message, y compris les FMS de la FESS (niveau métier) ainsi que certaines informations requises par le contrôle des flux de l'application afin de mettre en oeuvre le protocole de coordination.

Enfin, le DDNEA précise comment l'interface au bus de service du domaine commun EMCS définie dans les TESS [A11] peut être mise en place, afin d'assurer la *chorégraphie des processus métier* et le *contrôle des flux de l'application*. Bien que l'*orchestration des processus métier* soit principalement décrite dans la FESS [A1] au moyen des différents diagrammes de flux métier et diagrammes de transition d'états (State Transition Diagrams ou STD), le DDNEA inclut également certaines informations relatives à ce concept d'interface au bus de service du domaine commun EMCS. Les concepts d'*orchestration des processus métier*, de *chorégraphie des processus métier* et de *contrôle des flux de l'application* sont définis dans les TESS [A11].

En ce qui concerne l'*orchestration des processus métier*, la Section III-Métier central et la Section V-Services centraux contiennent des informations qui décrivent quelles sont les opérations qui devraient être effectuées dans chaque domaine pour chaque scénario particulier (scénario de base, changement de destination, etc.). La mise en oeuvre de l'*orchestration des processus métier* relève, quant à elle, de chaque pays puisqu'elle est liée à la manière dont la NDEA gèrera les échanges à l'intérieur de son domaine. La NDEA devra néanmoins se conformer aux opérations et protocoles d'échange de message définis dans les Sections mentionnées ci-dessus.

Les concepts de *chorégraphie des processus métier* et de *contrôle des flux de l'application* sont intégralement décrits dans le corps du DDNEA: à la Section III-Métier central, à la Section V-Services centraux, à la Section VIII-Principes de conception (VIII.I.3-Gestion des exceptions) et à la Section VI-Administration système. La FESS définit les processus métier EMCS. Le contrôle et la coordination de ces processus sont décrits dans les sections mentionnées ci-dessus.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

La Section III et la Section V indiquent clairement quel type de validation (métier ou technique) doit être effectuée pour chaque cas. Dans le cas d'une validation métier, l'appendice D définit les règles et les conditions qui doivent être respectées afin de garantir la validité métier du message. Dans le cas d'une validation technique, l'appendice H spécifie les XSD à partir desquels les messages doivent être vérifiés pour garantir leur validité technique.

Le DDNEA met par ailleurs en place le *protocole de coordination* et la *gestion des exceptions* du *contrôle des flux de l'application* en décrivant des protocoles d'échange de message pour les cas exceptionnels, en définissant les « state-machines » pour chaque lieu (Départ ou Destination), les différents rapports d'erreur en fonction du type d'erreur (de format ou fonctionnelle) et de certaines contraintes de conception et enfin en spécifiant le transport de messages via CCN/CSI (Section III, Section V, Section VIII et Section X). En dernier lieu, les activités d'enregistrement des données qui devraient être réalisées par les NDEA sont décrites à la Section VI.

En dehors des questions techniques, le DDNEA traite également différents aspects liés à la sécurité tels qu'ils sont détaillés dans les SESS [A10]. La sécurité est ainsi définie soit au niveau du transport, soit au niveau du message, en fonction du mécanisme de transport utilisé (CCN/CSI, SOAP/HTTP, etc.). Dans la Section X, la Section XI et la Section XII, des informations plus détaillées sur chaque mécanisme de transport peuvent être consultées.

Enfin, l'en-tête des messages figurant à l'appendice D a été défini afin de contenir toutes les informations requises par le *protocole de coordination*, la *gestion des exceptions* et le *contrôle des flux de l'application* dans leur ensemble (aspect sécurité compris).

PORTEE DU DOCUMENT

Le DDNEA se limite aux échanges d'information électronique au sein d'EMCS.

Cette version du DDNEA s'applique à la phase 3 d'EMCS. Les fonctionnalités concernées par cette phase sont définies dans le champ d'application d'EMCS, phase 3 [A2]. Le DDNEA est d'autre part conforme aux documents applicables figurant dans le tableau 1.

PUBLIC VISE

Le public visé par ce document comprend:

- Les personnes responsables des spécifications fonctionnelles d'EMCS.
- Les personnes responsables du développement de logiciels dans le cadre d'EMCS.
- Les personnes responsables de la définition de tests pour EMCS.
- Tous les collaborateurs des fournisseurs de services impliqués dans les projets CCN/CSI qui sont responsables de la prestation des services requis par EMCS.
- Tous les autres organismes autorisés concernés par EMCS, y compris le comité des accises, l'ECWP, le comité directeur ECP et les organisations professionnelles d'opérateurs économiques.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

Le public visé est supposé avoir une bonne compréhension de la terminologie et des concepts informatiques utilisés dans ce document. Il est également supposé avoir une bonne connaissance de la FESS [A1], du document spécifiant la portée d'EMCS [A2], des TESS [A11] et des SESS [A10].

STRUCTURE DU DOCUMENT

Ce document est structuré en sections et appendices. Les sections sont par la suite sous-divisées en chapitres. Sections sélectionnées sont sous-divisées en sous-sections. Les sections, sous-sections, chapitres et appendices sont résumés ci-dessous.

LA SECTION I - INFORMATIONS GÉNÉRALES comprend les chapitres suivants:

- Le Chapitre I.I.1 décrit les **relations entre ce document et les autres documents EMCS de référence**. Il définit un certain nombre de dépendances avec ces documents et détermine la validité d'application de ceux-ci. Il précise également quelles sont les normes dont il faut tenir compte au cours du développement et de la vérification de chaque application EMCS.
- Le Chapitre I.I.2 contient les **définitions** de termes employés dans ce document (terminologie, acronymes et abréviations).
- Le Chapitre I.I.3 décrit l'**objectif** et la **portée** du DDNEA, le **public visé**, la **structure interne** du document ainsi que des **informations de service** propres au document.

LA SECTION II - CHAMP D'APPLICATION DU DÉVELOPPEMENT passe en revue les éléments à développer dans les applications prévues en phase 3 d'EMCS. Les annexes A de la phase 3 d'EMCS qui correspondent à cet aspect complètent cette section.

Les sections suivantes présentent une définition détaillée des **protocoles de message** qui doivent être pris en charge dans les différents processus métier impliqués dans la gestion des accises. Ces protocoles de message sont décrits par une série de **diagrammes de séquence** accompagnés de **diagrammes de transition d'états**. Chacune de ces sections traite un des aspects liés aux processus métier:

SECTION III – ACTIVITÉ PRINCIPALE décrit l'activité principale de la phase 3 de l'EMCS. Plus particulièrement, cette section est sous-divisée en chapitres comme suit:

- La Sous-Section III.I définit les **scénarios de circuit central** et les éventuels échanges de message impliqués;
- La Sous-Section III.II spécifie les **scénarios d'export** et les éventuels échanges de message impliqués;
- La Sous-Section III.III présente les **diagrammes d'état-transition** et plus spécifiquement les états valides et les états-transitions;
- La Sous-Section III.IV décrit les **temporisateurs fonctionnels**;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

- La Sous-Section III.VI spécifie le mécanisme permettant de **traiter l'exception à l'aide** de certains scénarios exceptionnels.

SECTION IV – SUIVI ET COLLABORATION décrit le cas d'utilisation pour la coopération administrative chez les MSA. Plus particulièrement, cette section est sous-divisée en chapitres comme suit:

- La Sous-Section IV.I définit **les scénarios de suivi** pour signaler les événements et contrôles ayant lieu durant les mouvements d'accises;
- La Sous-Section IV.II spécifie **les scénarios de coopération administrative** qui décrivent les éventuels échanges de message concernant les informations spontanées et les demandes d'assistance;
- La Sous-Section IV.III présente **l'accès à l'historique des scénarios d'échange de message** concernant les mouvements pouvant être utilisés par les MSA afin d'accéder aux informations archivés relatives aux mouvements non disponibles en ligne;
- IV.II.3 et IV.III.2 décrit **les états-transitions se produisant** lors de l'exécution de la *coopération administrative* et l'accès à *l'historique des scénarios concernant les mouvements*.

LA SECTION V - SERVICES CENTRAUX traite avec les protocoles d'échange de message entre les NDEAs et la commune central du domaine des applications de services, à savoir le système d'échange de données relatives aux accises (SEED) et les services centraux / Système d'information de gestion pour l'accise (CS/MISE). Cette section est subdivisée comme suit:

- La Sous-Section V.I - **SEED** décrit le système d'échange de données relatives aux accises assurant la gestion et de diffusion d'informations sur les opérateurs économiques et les données de référence;
- La Sous-Section V.II - **CS/MISE** décrit les Services centraux / Système d'information de gestion pour fournir la surveillance d'accise et les services de rapports sur les opérations de l'EMCS, qui comprend la collecte, la distribution et la publication de statistiques sur les entreprises EMCS et technique, en recueillant et en publiant des informations sur les mouvements (ARC suivi), et la surveillance de la disponibilité de toutes les applications MSA (surveillance de la disponibilité).

LA SECTION VI - ADMINISTRATION SYSTÈME porte sur des questions telles que l'enregistrement et le traçage des données et sur toutes les autres fonctions d'administration à prévoir.

LA SECTION VII - STRUCTURE TECHNIQUE DU MESSAGE définit la structure technique détaillée des échanges d'information dans EMCS. Pour des raisons techniques, le format technique du message diffère parfois du format logique défini dans la FESS. Cette section est à son tour subdivisée comme suit:

- Le Chapitre VII.I.1 introduit le contenu de la Section VII.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

- Le Chapitre VII.I.2 présente le **dictionnaire de données**. Il définit certains des éléments qui composent un message comme les éléments de données, les groupes de données et les listes de codes (séries de valeurs discrètes). L'appendice G, l'appendice F et l'appendice B complètent ce chapitre.
- Le Chapitre VII.I.3 détaille la **structure technique du message** (Technical Message Structure ou TMS) pour les différents échanges d'information. Le détail des TMS correspondant à chaque message figure à l'appendice D. Dans ce chapitre, il sera uniquement expliqué comment l'appendice D doit être interprété et utilisé.
- Le Chapitre VII.I.4 décrit le 'Message Header' utilisé dans les Echanges d'Information
- Le Chapitre VII.I.5 étudie la question de la **cohérence**. Il définit les documents relatifs à la gestion des accises avec lesquels ce DDNEA doit concorder (tel que « FESS » [A1]) et explique comment garantir cette cohérence au cours de la définition des TMS.

La SECTION VIII - Principes de conception explique comment doit être mis en place le système défini dans les sections précédentes. Tout échange d'information doit avant tout être converti au format XML et transmis par l'intermédiaire d'une des trois plateformes de communication (CCN/CSI et SOAP/HTTP). Cette section fixe un certain nombre de principes communs à tous les mécanismes de transport:

- Le Chapitre VIII.I.1 explique quelle est l'**approche** globale.
- Le Chapitre VIII.I.2 est consacré à l'utilisation des **jeux de caractères** et des **conventions portant sur les éléments de données**.
- Le Chapitre VIII.I.3 définit les détails techniques de la **gestion des exceptions** en phase 3 d'EMCS.
- Le Chapitre VIII.I.4 définit les **contraintes** (toutes les restrictions applicables au développement d'EMCS).

LA SECTION IX - FORMATAGE XML définit la façon dont les messages doivent être convertis au format XML. Cette section est structurée comme suit:

- Le Chapitre IX.I.1 définit les **conventions XML** établies pour EMCS.
- Le Chapitre IX.I.2 indique quels sont les **jeux de caractères** qui doivent être pris en charge par les NDEA.
- Le Chapitre IX.I.3 est consacré au formatage XML des **échanges d'information**. L'appendice E complète ce chapitre.
- Le Chapitre IX.I.4 est consacré aux **schémas XML** des messages EMCS. L'appendice H complète ce chapitre.

LA SECTION X - TRANSPORT DE MESSAGES VIA CCN/CSI définit la façon dont les messages doivent être transportés par l'intermédiaire de la plateforme de communication CCN/CSI. Cette section est subdivisée comme suit:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

- Le Chapitre X.I.1 définit les **hypothèses architecturales** formulées pour permettre le transport de messages via CCN/CSI et précise à quel endroit les références au CCN/CSI peuvent être consultées.
- Le Chapitre X.I.2 présente les **éléments CCN/CSI obligatoires** qui assureront toute la communication d'une passerelle CCN à l'autre.
- Le Chapitre X.I.3 présente les **éléments CCN/CSI recommandés** pour pouvoir envoyer et recevoir des messages.
- Le Chapitre X.I.4 définit les **informations de configuration** nécessaires aux passerelles CCN.

LA SECTION XI - TRANSPORT DE MESSAGES VIA SOAP/HTTP définit le transport de messages via SOAP/HTTP qui peut être utilisé par une NDEA pour faire exécuter les services web des services centraux du domaine commun. Cette section est divisée comme suit:

- La Sous-Section XI.I - **Topologie** décrit la **topologie** du transport HTTP par l'intermédiaire de CCN, spécifie les **URI** des services web des services centraux du domaine commun et définit la syntaxe du **chemin relatif du service web**.
- La Sous-Section XI.II - **Configuration CCN** décrit la **configuration CCN/HTTP** permettant de rendre accessible les services web des services centraux.
- La Sous-Section XI.III - **Normes Pour Services Web** fournit des informations relatives aux **normes établies pour les services web** SOAP, HTTP, XML-RPC, WSDL et WS-Security.
- La Sous-Section XI.IV - **Utilisation Recommandée** explique quelles sont les conditions que doit remplir une NDEA qui communique avec les services web par l'intermédiaire de **SOAP/HTTP**.
- La Sous-Section XI.V - **Transactions Des Services Web** fournit des informations relatives aux **transactions réalisées par les services web** et décrit ce qu'elles impliquent.
- La Sous-Section XI.VI - **Sécurité Des Services Web** définit la **sécurité des services web** appliquée au niveau transport et au niveau message.
- La Sous-Section XI.VII - **Authentification Http De L'utilisateur CCN** précise le mécanisme d'**authentification** à utiliser dans CCN/HTTP.

LA SECTION XII - SERVICES INTRANET D'AUTHEMIFICATION DES APPLICATIONS décrit les services qui déploient des interfaces HTTP permettant aux applications web de se connecter (ccnServerLogin) et se déconnecter (ccnServerLogout) à partir de CCN.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

APPENDICES

- L'appendice A présente tous les messages entrant dans le champ d'application du DDNEA en phase 3 d'EMCS.
- L'appendice B contient une définition de toutes les listes de codes utilisées pour EMCS qui sont applicables en phase 3.
- L'appendice C présente ce en quoi les différents groupes de données et les différents éléments de données sont en corrélation avec les messages.
- L'appendice D contient la définition de tous les messages utilisés en phase 3 d'EMCS.
- L'appendice E contient le mappage XML de tous les éléments de données et groupes de données présents dans les messages EMCS.
- L'appendice F et l'appendice G contiennent un dictionnaire de données qui répertorie tous les éléments (éléments de données et groupes de données) utilisés pour composer ces messages.
- L'appendice H présente les schémas XML correspondant à tous les messages utilisés dans EMCS.
- L'appendice I définit les fichiers WSDL des services web mis en place par les services centraux.
- L'appendice J contient le catalogue des Business Rules (BRS) qui ont été identifiés dans les processus d'affaires élémentaires de la FESS Section II: CORE BUSINESS.
- L'appendice K contient, par ordre alphabétique, la liste complète des règles et conditions et les voies IE qui lui sont associées.

INFORMATIONS DE SERVICE PROPRES AU DOCUMENT

Les différentes parties qui constituent le DDNEA seront soumises, une par une, à un contrôle des configurations et des versions. Chacune des composantes peut être mise à jour et livrée séparément.

Ce document sera tenu à jour. La DG Fiscalité et union douanière définira les différentes livraisons et en établira le calendrier.

Des commentaires concernant ce document peuvent être transmis soit par le biais de revues officielles soit en appelant le service d'assistance central d'EMCS.

Les erreurs identifiées dans ce DDNEA seront répertoriées sous forme de liste d'erreurs connues (Known Error List ou KEL) mise à disposition par le service d'assistance central d'EMCS.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Résumé	

Quand une partie de ce document sera mentionnée, la référence indiquée renverra soit à une section entière, soit à un chapitre entier (à l'intérieur d'une section), soit à un paragraphe (pour toute autre subdivision).

Ce document sera envoyé sous forme de fichiers Word portant les noms suivants:

- **ECP3- FITSDEV3-SC01-DDNEA_P3-Vy.zz-EN.doc**, y et zz indiquant le numéro de la version et celui de la révision.

Toutes les annexes de la phase 3 d'EMCS se présenteront de la façon suivante:

- **ECP3- FITSDEV3-SC01C03-DDNEA_P3_App_X_yzz-EN.DDD**, X indiquant le nom de l'annexe, y et zz le numéro de la version et celui de la révision et DDD le type de document (PDF pour un fichier Adobe Acrobat 5, DOC pour MS Word, MDB pour MS Access).

Il est à noter que les annexes ne sont pas modifiées séparément. Cela signifie qu'elles porteront le même numéro de version que le DDNEA (**yzz**).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

ZUSAMMENFASSUNG

ZIELSETZUNG DIESES DOKUMENTS

Dieses Dokument ist das **Dokument für den Entwurf Nationaler Verbrauchsteueranwendungen** (DDNEA - Design Document for National Excise Applications). Es spezifiziert die Anforderungen, denen jede Verbrauchsteuerkontrollanwendung entsprechen muss.

Dies gilt für die von einer Mitgliedstaatsverwaltung (MSA - Member State Administration) national entwickelten Verbrauchsteueranwendungen (NDEA - Nationally Developed Excise Applications) und für die vom Zentralen Projektteam (CPT - Central Project Team) zentral entwickelten Verbrauchsteueranwendungen (CDEA - Centrally Developed Excise Applications).

Dieses Dokument **gilt für jede national entwickelte Verbrauchsteueranwendung (NDEAs)** und für die **zentral entwickelten Verbrauchsteueranwendungen (CDEAs)** und ist als obligatorisches Dokument für die Implementierungs- und Verifizierungsaktivitäten zu behandeln.

Dieses Dokument erfüllt einen zweifachen Zweck:

- Es stellt klar, **was** entwickelt werden muss. Dies soll erreicht werden, indem die Abfolgen von Informationsaustauschen, die unterstützt werden, als eine Anzahl von Protokollen für den Nachrichtenaustausch sowie außerdem die detaillierte Struktur und Erstellungsregeln für diese Informationsaustausche spezifiziert werden.
- Es definiert, **wie** die Informationsaustausche stattzufinden haben. Im Wesentlichen muss jeder Informationsaustausch als eine XML-Darstellung formatiert (oder dargestellt) werden, und diese formatierte Nachricht muss zwischen den Verbrauchsteueranwendungen transportiert werden.

Für die NEAs sind die Informationsaustausche in dem Gemeinsamen Bereich (zwischen den Mitgliedstaatsverwaltungen) und in dem Externen Bereich (zwischen Mitgliedstaatsverwaltungen und Wirtschaftsbeteiligten) vorgesehen.

Diese DDNEA definiert in erster Linie Aspekte für die *Anwendungsebene* und die *Infrastrukturebene* der Architektur des Gemeinsamen EMCS-Bereichs [A11]. Konkret enthält das DDNEA alle Informationen, die für eine Implementierung dieser Architekturebenen des Gemeinsamen EMCS-Bereichs durch die NEAs erforderlich sind. Die *Geschäftsebene* wird von der FESS [A1] definiert, da die erwarteten „Dienstleistungen“ für das EMCS darin beschrieben werden. Dennoch berücksichtigte die DDNEA für die Definition der Aspekte der *Anwendungsebene* diese Ebene und die FESS [A1], da die Anwendungsflusskontrolle auf der Grundlage der „Dienstleistungen“ definiert ist, die vom EMCS für die Phase 3 erwartet werden.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

Was die EMCS Geschäfts-Kommunikationskanäle (BCC - Business Communication Channels) betrifft, so ist diese DDNEA für die folgenden relevant: BCC2, BCC7, BCC6, BCC9, BCC10, BCC11 und BCC12. Die EMCS Infrastruktur-Kommunikationskanäle (ICC - Infrastructure Communication Channels), für die die DDNEA relevant ist, sind folgende: ICC4, ICC5, ICC6, ICC11, ICC15, ICC16 und ICC17.

Die DDNEA bietet Informationen darüber, wie die NEAs mittels Kommunikation über das Gemeinsame Bereichs-Relais entweder die CCN/CSI-Dienste oder die CCN-Intranet-Dienste benutzen. Konkret wird in Abschnitt X-Transport von Nachrichten über CCN/CSI spezifiziert, wie die Nachrichten unter Berücksichtigung von BCC2, BCC7, BCC6, BCC10, BCC12 und ihrer Zerlegung in ICC4, ICC5, ICC15 und ICC16 über CCN/CSI transportiert werden müssen. Der Abschnitt XI-Transport von Nachrichten definiert den Transport von Nachrichten über HTTP. Diese Abschnitte spezifizieren, wie eine NEA die Webservices der Zentralen Dienstleistungen des Gemeinsamen Bereichs aufrufen kann oder Informationen entweder über eine manuelle (Webbrowser) oder programmatische Schnittstelle austauschen kann, wobei BCC7, BCC6, BCC9, BCC10, BCC11 und BCC12 und ihre Zerlegung in ICC6, ICC11 und ICC17 berücksichtigt werden.

Darüber hinaus definiert die DDNEA auch Format und Struktur der EMCS-Nachricht. Der Abschnitt IX- XML-Formatierung beschreibt, wie die EMCS-Nachrichten in ein XML-Format formatiert werden müssen. Außerdem werden in Abschnitt VIII-Entwurfs-Prinzipien einige allgemeine Prinzipien dargestellt, die unabhängig vom Transportmechanismus (CCN/CSI und SOAP/HTTP) Anwendung finden. Schließlich wird in Abschnitt VII-Technische Struktur der Nachrichten die Struktur einer EMCS-Nachricht definiert. In diesem Abschnitt wird gezeigt, dass das DDNEA die EMCS-Nachrichtenstruktur sowohl der *Geschäftsebene* als auch der *Anwendungsebene* berücksichtigt. Anhang D der DDNEA präsentiert im Detail die technische Nachrichtenstruktur, einschließlich des FMS der FESS (Geschäftsebene) sowie einige der für die Anwendungsflusskontrolle erforderlichen Informationen, die eine Implementierung des Koordinationsprotokolls ermöglichen.

Schließlich spezifiziert die DDNEA, wie die in der TESS definierte Servicebus-Schnittstelle des Gemeinsamen Bereichs [A11] implementiert werden kann, um die *Geschäftsvorgangschoreographie* und die *Anwendungsflusskontrolle* zu gewährleisten. Auch wenn die *Geschäftsvorgangsinstrumentalisierung* vor allem in FESS [A1] in den Geschäftsstromdiagrammen und den STDs präsentiert wird, enthält das DDNEA auch einige Informationen für das Konzept der Servicebus-Schnittstelle des Gemeinsamen EMCS-Bereichs. Die Konzepte *Geschäftsvorgangsinstrumentalisierung*, *Geschäftsvorgangschoreographie* und *Anwendungsflusskontrolle* werden in TESS [A11] definiert.

Zur *Geschäftsvorgangsinstrumentalisierung* finden sich in Abschnitt III-Kerngeschäft und Abschnitt V-Zentrale Dienstleistungen, in denen beschrieben werden, welche Operationen in den einzelnen Bereichen für jedes einzelne Szenarium durchgeführt werden sollten (GrundszENARIO, Änderung des Zielortes usw.). Die Implementierung der *Geschäftsvorgangsinstrumentalisierung* ist jedoch ein nationales Anliegen, da sie die Frage betrifft, wie die NDEA bereichsinterne Austausche handhabt. Die NDEA sollte auf jeden Fall zumindest an den Nachrichtenaustauschprotokollen und Operationen, die in den zuvor beschriebenen Abschnitten definiert wurden, ausgerichtet sein.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

Die Konzepte *Geschäftsvorgangschoreographie* und *Anwendungsflusskontrolle* werden in der DDNEA in den Abschnitten III-Kerngeschäft, Abschnitt V-Zentrale Dienstleistungen, Abschnitt VIII-Gestaltungs-Prinzipien (VIII.I.3 - Umgang mit Ausnahmen) und Abschnitt VI-Systemverwaltung. Die FESS definiert die EMCS-Geschäftsvorgänge. Die Kontrolle und Koordinierung dieser Vorgänge wird in den zuvor angeführten Abschnitten beschrieben.

Abschnitt III und Abschnitt V legen klar und deutlich dar, welche Art von Validierungen (geschäftlich oder technisch) je nach Fall durchzuführen sind. Im Falle einer Geschäftsvalidierung definiert Anhang D, welche Regeln und Bedingungen für eine Gewährleistung der fachliche Gültigkeit der Nachricht befolgt werden müssen. Was die technische Validierung betrifft, so spezifiziert Anhang H die XSDs, auf deren Grundlage eine Nachricht für die Gewährleistung ihrer technischen Validität überprüft werden soll.

Darüber hinaus implementiert die DDNEA das *Koordinierungsprotokoll* und den *Umgang mit Ausnahmen* der *Anwendungsflusskontrolle* mit einer Beschreibung der Nachrichtenaustauschprotokolle unter Berücksichtigung von Ausnahmefällen, der Definition der State-Machines pro Ort (Versand- oder Bestimmungsort), der Definition von Fehlermeldungen je nach Fehlerart (Formatfehler oder funktioneller Fehler) sowie einiger Einschränkungen in der Gestaltung und schließlich mit der Spezifikation für den Nachrichtentransport über CCN/CSI (Abschnitt III, Abschnitt V, Abschnitt VIII und Abschnitt X). Schließlich werden in Abschnitt VI die von den NDEAs durchzuführenden Log-Aktivitäten beschrieben.

Abgesehen von technischen Aspekten geht die DDNEA auch auf Sicherheitsaspekte, wie in der SESS [A10] definiert, ein. Folglich wird die Sicherheit je nach Transportmechanismus (CCN/CSI, SOAP/HTTP, etc.) entweder auf Transportebene oder auf Nachrichtenebene definiert. Weitere Informationen zu den einzelnen Transportmechanismen finden sich in Abschnitt X, Abschnitt XI und Abschnitt XII.

Als eine Schlussfolgerung wurde die Kopfzeile der Nachrichten in Anhang D so definiert, dass sie all die erforderlichen Informationen zum *Koordinierungsprotokoll*, zum *Umgang mit Ausnahmen* und zur *Anwendungsflusskontrolle* allgemein (einschließlich Sicherheit) enthält.

GELTUNGSBEREICH DES DOKUMENTS

Die DDNEA beschränkt sich auf den elektronischen Informationsaustausch im Rahmen von EMCS.

Diese Version der DDNEA findet auf die EMCS-Phase 3 Anwendung. Die für diese Phase relevante Funktionalität wird in der Zielsetzung des EMCS für Phase 3 [A2] definiert. Darüber hinaus ist das DDNEA mit den in Tabelle 1 angeführten anwendbaren Dokumenten verbunden.

ZIELGRUPPE

Zur Zielgruppe dieses Dokuments gehören:

- Jede für die funktionellen Spezifikationen des EMCS zuständige Person.
- Jede für die Entwicklung von Software im Rahmen des EMCS zuständige Person.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

- Jede für die Definition von Tests für das EMCS zuständige Person.
- Jeder der betroffenen Leistungsträger im Rahmen von CCN/CSI-Projekten, der für die Lieferung von für das EMCS erforderlichen Dienstleistungen verantwortlich ist.
- Jede sonstige Einrichtung, für die das EMCS relevant ist, einschließlich dem Verbrauchsteuerausschuss, dem ECWP, dem ECP-Lenkungsausschuss und der Wirtschaftsverbände.

Vom Zielpublikum wird ein gutes Verständnis der in diesem Dokument verwendeten IT-Begriffe und Terminologie vorausgesetzt. Außerdem wird davon ausgegangen, dass sie mit der FESS [A1], dem Dokument „Ziel des EMCS“ [A2], der TESS [A11] und der SESS [A10] vertraut sind.

STRUKTUR DES DOKUMENTS

Dieses Dokument ist strukturiert in Abschnitte und Anhänge. Die Abschnitte sind weiter in Kapitel unterteilt. Ausgewählte Abschnitte sind weiter in sub-Abschnitte unterteilt. Die Abschnitte, sub-Abschnitte, Kapitel und Anhänge sind unten:

ABSCHNITT I - ALLGEMEINE INFORMATIONEN umfasst die folgenden Kapitel:

- Kapitel I.I.1 beschreibt die **Beziehung dieses Dokuments mit anderen EMCS-Basisdokumenten**. Es definiert die Relationen zu diesen Dokumenten und verweist auf die Anwendbarkeit dieser Dokumente. Außerdem spezifiziert es, welche Normen bei der Entwicklung und Verifizierung einer EMCS-Anwendung berücksichtigt werden müssen.
- Kapitel I.I.2 enthält die in diesem Dokument verwendeten **Definitionen** (Terminologie, Akronyme und Abkürzungen).
- Kapitel I.I.3 beschreibt den **Zweck** und die **Zielsetzung** des DDNEA, die **Zielgruppe**, die **interne Struktur** des Dokuments und sowie außerdem einige **Dokument-Dienstleistungsinformationen**.

ABSCHNITT II - Entwicklungsumfangbehandelt die Funktionen, die für die Anwendungen der EMCS-Phase 3 entwickelt werden müssen. Zu diesem Abschnitt gehören die einschlägigen Anhänge A für die EMCS-Phase 3.

Die anschließenden Abschnitte enthalten eine detaillierte Definition der **Nachrichtenprotokolle**, die von den verschiedenen Verbrauchsteuer-Geschäftsvorgängen unterstützt werden müssen. Diese Nachrichtenprotokolle werden von einer Sammlung von **Zeitfolgediagrammen** beschrieben und durch Zustandsübergangsdigramme unterstützt. Jeder Abschnitt behandelt einen der folgenden Geschäftsvorgangsbereiche:

ABSCHNITT III - KERNGESCHÄFT beschreibt das Kerngeschäft für EMCS-Phase 3. Konkret ist dieser Abschnitt in die folgenden Kapitel untergliedert:

- Unterabschnitt III.I.1 definiert die **Zentralen Kreislaufszzenarien** und den möglichen Nachrichtenaustausch, die beziehen sind;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

- Unterabschnitt III.II spezifiziert die **Export-Drehbücher (Szenarios)** und den möglichen Nachrichtenaustausch, die beziehen sind;
- Unterabschnitt III.III präsentiert der **State-Transition-Diagramme** und spezifisch die gültigen Staaten und Zustandübergänge;
- Unterabschnitt III.IV beschreibt die **Funktionellen Timer**;
- Unterabschnitt III.V spezifiziert die **Ausnahmebehandlung Mechanismus** mit einigen außergewöhnlichen Szenarien.

ABSCHNITT IV - ANSCHLUSS UND KOLLABORATION beschreibt den Gebrauchfall für die administrative kollaboration zwischen MSAs. Konkret ist dieser Abschnitt in die folgenden Kapitel untergliedert:

- Unterabschnitt IV.I definiert definieren die **Anschlussdrehbücher** für Berichtsergebnisse und Kontrollen, die während der Verbrauchsteuerbewegungen auftreten;
- Unterabschnitt IV.II spezifiziert die **Verwaltungsmitarbeitdrehbücher**, der den möglichen Mitteilungsaustausch der spontanen Informationen beschreibt und für Unterstützung verlangt;
- Unterabschnitt IV.III präsentiert Zugang zur **Geschichte der Bewegungsmitteilung-Austauschdrehbücher**, die durch den MSAs benutzt werden können, um zurückzugreifen, archivierten Informationen für vorhandenes on-line der Bewegungen nicht;
- IV.II.3 und IV.III.1.6 beschreibt die **Zustand-Übergänge**, die während der Durchführung der *Verwaltungsmitarbeit* und des *Zugangs zur Geschichte der Bewegungsmitteilung* auftreten.

ABSCHNITT V - ZENTRALE DIENSTLEISTUNGEN beschäftigt sich mit dem Austausch von Nachrichten-Protokolle zwischen den NDEAs und der Gemeinsamen Domain Central Services-Anwendungen, nämlich das System für den Austausch von Excise Data (SEED) und die Central Services / Management Information System für Verbrauchsteuern (CS/MISE). Dieser Abschnitt ist wie folgt unterteilt:

- Unterabschnitt V.I - **SEED** beschreibt das System für den Austausch von Verbrauchsteuerdaten Bereitstellung von Management und Verbreitung von Dienstleistungen von Informationen über die Wirtschaftsbeteiligten und die Reference Data;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

- Unterabschnitt V.II - **CS/MISE** beschreibt die Central Services / Management Information System für Verbrauchsteuern bietet die Überwachung und das Reporting Services auf den Betrieb von EMCS, das Sammeln, Verteilen und Veröffentlichen von EMCS geschäftlichen und technischen Statistiken enthält, durch das Sammeln und Veröffentlichung von Informationen über Bewegungen (ARC Follow-up) sowie die Überwachung der Verfügbarkeit aller MSA-Anwendungen (Überwachung der Verfügbarkeit).

ABSCHNITT VI - SYSTEMVERWALTUNG geht auf Themen wie Protokollierung und Rückverfolgung sowie andere vorgesehene Verwaltungsfunktionen ein.

ABSCHNITT VII - TECHNISCHE NACHRICHTSSTRUKTUR definiert im Detail die technische Struktur des EMCS-Informationsaustausches. Aus technischen Gründen unterscheidet sich das technische Nachrichtenformat gelegentlich von dem in der FESS beschriebenen logischen Format. Dieser Abschnitt ist wie folgt unterteilt:

- Kapitel VII.I.1 stellt den Inhalt der Sektion VI vor.
- Kapitel VII.I.2 führt das **Datenbeschreibungsverzeichnis** ein. Dieses definiert eine Reihe von Objekten, die eine Nachricht bildet, wie Datenobjekte, Datengruppen und Codelisten (Satz an diskreten Werten). Dieses Kapitel wird durch Anhang G, Anhang F und Anhang B ergänzt.
- Kapitel VII.I.3 präsentiert im Detail die **Technische Nachrichtenstruktur** (TMS - Technical Message Structure) für die verschiedenen Informationsaustausche. Eine detaillierte TMS für alle Nachrichten findet sich in Anhang D. Dieses Kapitel enthält lediglich Erklärungen zum Verständnis und zur Verwendung von Anhang D.
- Kapitel VII.I.4 präsentiert die Kopfzeile der Nachrichten verwendet in Informationsaustausch.
- Kapitel VII.I.5 diskutiert über den Aspekt der **Übereinstimmung**. Es definiert, mit welchen Verbrauchsteuerdokumenten diese DDNEA konsistent sein muss („FESS“ [A1]) und erläutert, wie diese Übereinstimmung während der TMS-Definition gewährleistet wurde.

ABSCHNITT VIII - Ausgestaltungsprinzipien erklärt, wie das in den vorstehenden Abschnitten definierte System eingerichtet werden muss. Im Wesentlichen muss jeder Informationsaustausch in XML formatiert sein und über eine der drei Kommunikationsplattformen (CCN/CSI und SOAP/HTTP) übertragen werden. Dieser Abschnitt führt eine Reihe von Prinzipien an, die unabhängig vom Transportmechanismus gelten:

- Kapitel VIII.I.1 diskutiert über den **Gesamtansatz**.
- Kapitel VIII.I.2 diskutiert über die Behandlung von **Zeichensätzen** und **Datenobjektkonventionen**.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

- Kapitel VIII.I.3 definiert die technischen Details für den **Umgang mit Ausnahmen** in der EMCS-Phase 3.
- Kapitel VIII.I.4 definiert **Einschränkungen** (sämtliche Beschränkungen, die die Entwicklung des EMCS betreffen).

ABSCHNITT IX - XML-FORMATIERUNG definiert die Formatierung der Nachrichten in einem XML-Format. Dieser Abschnitt ist wie folgt aufgebaut:

- Kapitel IX.I.1 definiert die für das EMCS geltenden **XML-Konventionen**.
- Kapitel IX.I.2 spezifiziert die **Zeichensätze**, die von den NDEAs unterstützt werden müssen.
- Kapitel IX.I.3 behandelt die XML-Formatierung der **Informationsaustausche**. Dieses Kapitel wird durch Anhang E ergänzt.
- Kapitel IX.I.4 behandelt die **XML-Schemata** von EMCS-Nachrichten. Dieses Kapitel wird durch Anhang H ergänzt.

ABSCHNITT X - TRANSPORT VON NACHRICHTEN ÜBER CCN/CSI definiert, wie Nachrichten über die CCN/CSI-Kommunikationsplattform transportiert werden. Dieser Abschnitt ist wie folgt untergliedert:

- Kapitel X.I.1 definiert die **architektonischen Annahmen** für den Nachrichtentransport über CCN/CSI und beschreibt genau, wo Hinweise auf CCN/CSI gefunden werden können.
- Kapitel X.I.2 präsentiert die **obligatorischen CCN/CSI-Elemente**, die eine Endpunkt-Kommunikation zwischen zwei CCN-Portalen gewährleisten.
- Kapitel X.I.3 präsentiert die für das Versenden und Empfangen von Nachrichten **empfohlenen CCN/CSI-Elemente**.
- Kapitel X.I.4 definiert die für die CCN-Portale erforderliche **Konfigurationsinformation**.

ABSCHNITT XI - TRANSPORT VON NACHRICHTEN ÜBER SOAP/HTTP definiert den Transport von Nachrichten über SOAP/HTTP, mit dessen Hilfe eine NDEA den Gemeinsamen Bereich Zentrale Dienstleistungs-Webservices abrufen kann. Dieser Abschnitt ist wie folgt untergliedert:

- Unterabschnitt XI.I - **Topologie** beschreibt die **Topologie** des HTTP-Transports über CCN, spezifiziert die **URIs** für den Gemeinsamen Bereich Zentrale Dienstleistungs-Webservices und definiert die Syntax des **relativen Webservice-Pfades**.
- Unterabschnitt XI.II - **CCN-Konfiguration** beschreibt die **CCN/HTTP-Konfiguration**, die den Zugriff auf die Zentrale Dienstleistungs-Webservices gewährleistet.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

- Unterabschnitt XI.III - **Webservice-Normen** bietet Informationen zu den folgenden **Webservice-Normen**: SOAP, HTTP, XML-RPC, WSDL und WS-Security.
- Unterabschnitt XI.IV - **Empfohlene Verwendung** erklärt die Empfehlungen für ein NDEA, das über den **SOAP/HTTP** mit den Webservices kommuniziert.
- Unterabschnitt XI.V - **Webservice-Transaktionen** bietet Informationen und beschreibt die Implikation der **Webservice-Transaktionen**.
- Unterabschnitt XI.VI - **Webservice-Sicherheit** definiert die **Sicherheit der Webservices** auf Transport- und Nachrichtenebene.
- Unterabschnitt XI.VII - **Http-Zugriffsberechtigung der CCN-Benutzer** spezifiziert die **Zugriffsberechtigungs-** Mechanismen in CCN/HTTP.

ABSCHNITT XII - INTRANET-DIENSTLEISTUNGEN FÜR DIE ZUGRIFFSBERECHTIGUNG AUF DIE ANWENDUNG beschreibt die Dienstleistungen, die HTTP-Schnittstellen zu den Web-Anwendungen zum **Einloggen** (ccnServerLogin) und **Ausloggen** (ccnServerLogout) von CCN bieten.

ANHÄNGE

- Anhang A präsentiert alle Nachrichten, die unter die Zielsetzung des DDNEA für die EMCS-Phase 3 fallen.
- Anhang B enthält eine Definition aller für das EMCS verwendeten Codelisten, die auf die Phase 3 Anwendung finden.
- Anhang C stellt dar, wie verschiedene Datengruppen und Datenobjekte mit den Nachrichten korrelieren.
- Anhang D enthält eine Liste mit allen Nachrichten für die EMCS-Phase 3.
- Anhang E enthält das XML-Mapping aller Datenobjekte und Datengruppe der EMCS-Nachrichten.
- Anhang F und Anhang G enthalten ein Datenbeschreibungsverzeichnis für alle Elemente (Datenobjekte und Datengruppen), die für die Erstellung dieser Nachrichten verwendet werden.
- Anhang H enthält die XML-Schemata für alle im EMCS verwendeten Nachrichten.
- Anhang I definiert die WSDL-Dateien für die von den Zentralen Dienstleistungen erbrachten Webservices.
- Anhang J enthält den Katalog der Business Rules (BRS), die in den elementarer Geschäftsprozesse der FESS Section II: CORE BUSINESS.
- Anhang K enthält, in alphabetischer Reihenfolge, eine vollständige Liste der gültigen Regeln und Bedingungen und die IE Pfade die mit Ihnen verbunden sind.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Zusammenfassung	

DOKUMENTSERVICE-INFORMATIONEN

Die verschiedenen Teile des DDNEA werden einzeln einer Konfigurations- und Versionskontrolle unterzogen. Einzelne Teile können getrennt aktualisiert und bereitgestellt werden.

Dieses Dokument wird regelmäßig überarbeitet. Die GD Steuern und Zollunion definiert und plant die verschiedenen Lieferungen.

Anmerkungen zu diesem Dokument können entweder mittels organisierter Revisionen oder mittels Anrufen beim Zentralen EMCS-Helpdesk eingereicht werden.

Bekannte Fehler in diesem DDNEA werden im Format der Liste mit bekannten Fehlern (KEL - Known Error List) geführt, die im Zentralen EMCS-Helpdesk veröffentlicht ist.

Wenn auf einen Teil dieses Dokuments verwiesen wird, so erfolgt der Verweis immer auf einen gesamten Abschnitt oder ein gesamtes Kapitel (innerhalb eines Abschnitts) oder einen Absatz (für sonstige Unterteilungen).

Dieses Dokument wird als eine Word-Datei mit der folgenden Namenskonvention eingereicht werden:

- **ECP3- FITSDEV3-SC01-DDNEA_P3-Vy.zz-EN.doc**, wobei y und zz für die Versions- und Revisionsnummer stehen.

Alle anhänge zur EMCS-Phase 3 werden wie folgt geliefert:

- **ECP3- FITSDEV3-SC01-DDNEA_P3_App_X_yzz-EN.DDD**, wobei X für den Anhangnamen, y und zz für die Versions- und Revisionsnummer und DDD für die Dokumentenart (PDF für eine Adobe Acrobat 5-Datei, DOC für MS Word, MDB für MS Access) stehen.

Bitte beachten Sie, dass anhänge nicht eigenständig herausgegeben werden. Dies bedeutet, dass die anhänge immer die gleiche Versionsnummer wie das DDNEA (**yzz**) haben.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

TABLE OF CONTENTS

DOCUMENT HISTORY	2
EXECUTIVE SUMMARY	8
RESUME	16
ZUSAMMENFASSUNG	25
TABLE OF CONTENTS	34
LIST OF FIGURES	44
LIST OF TABLES	52
SECTION I GENERAL INFORMATION	54
<i>I.1.1 Applicable and Reference documents</i>	<i>54</i>
I.I.1.1 Applicable Documents and Standards	54
I.I.1.1.1 Documents	54
I.I.1.1.2 Standards	55
I.I.1.2 Reference Documents and Standards	56
I.I.1.2.1 Documents	56
<i>I.1.2 Definitions</i>	<i>56</i>
I.I.2.1 Definitions	56
I.I.2.2 Terminology	57
I.I.2.3 Acronyms and Abbreviations	58
<i>I.1.3 Symbolism and Conventions Used</i>	<i>60</i>
I.I.3.1 Time Sequence Diagrams	61
I.I.3.2 State Transition Diagrams	64
I.I.3.3 Data dictionary	65
I.I.3.3.1 Data Items.....	65
I.I.3.3.2 Data Groups	65
I.I.3.3.3 Codelists	65
SECTION II SCOPE OF DEVELOPMENT	66
<i>I.I.1 Information Exchange Overview for EMCS Phase 3</i>	<i>66</i>
<i>I.I.2 Information Exchange Map of EMCS Phase 3 for Core Business</i>	<i>67</i>
SECTION III CORE BUSINESS	68
SUB-SECTION III.I CENTRAL CIRCUIT SCENARIOS	68
<i>III.I.1 Submission and registration of an e-AD (UC2.01)</i>	<i>68</i>
III.I.1.1 Origin is tax warehouse	69
III.I.1.1.1 Origin is tax warehouse and the destination is known.....	69
III.I.1.1.2 Origin is tax warehouse and the destination is unknown.....	71
III.I.1.2 Origin is import.....	72
III.I.1.2.1 Origin is import and the e-AD is inconsistent with the import data	72
III.I.1.2.2 Origin is import and the destination is known	74
III.I.1.2.3 Origin is import and the destination is unknown	76
<i>III.I.2 Alert or rejection of an e-AD (UC2.07)</i>	<i>78</i>
III.I.2.1 e-AD alerted.....	78
III.I.2.2 e-AD rejected	80
<i>III.I.3 Cancellation of an e-AD by the Consignor (UC2.10)</i>	<i>81</i>

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

<i>III.I.4 Submission of Report of Receipt (UC2.06)</i>	84
III.I.4.1 Delivery Accepted	85
III.I.4.2 Delivery Refused	86
III.I.4.3 Delivery Partially Refused.....	88
<i>III.I.5 Change of Destination (UC2.05)</i>	89
III.I.5.1 Change MS of Destination.....	90
III.I.5.2 Change MS of Destination where the consignor diverts a movement back to a (former) MSA of Destination.....	93
III.I.5.3 Return to the tax warehouse of dispatch.....	96
III.I.5.4 Change of Consignee (not the MS of Destination).....	98
III.I.5.5 Change of Place of Delivery	101
<i>III.I.6 Splitting of Consignment (UC2.36)</i>	102
<i>III.I.7 Reminder at expiry of time limit for Report of Receipt (UC2.33)</i>	106
III.I.7.1 Status Request/Response after the TIM_EAD timer expiration.....	107
III.I.7.2 Submission of explanations on delay for delivery	109
<i>III.I.8 Reminder at expiry time for change of destination (UC2.17)</i>	111
<i>III.I.9 Reminder at expiry of time limit for update of destination fields (UC2.34)</i>	112
III.I.9.1 Expiration of the time limit to update the destination fields.....	112
III.I.9.2 Expiration of the time limit to update the destination fields with Consignor explanations.....	113
<i>III.I.10 Post-delivery processing (UC2.12)</i>	114
<i>III.I.11 Information on intended claims (UC2.13)</i>	116
III.I.11.1 Duties claimed by the MSA of Destination	116
III.I.11.2 Duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination	117
<i>III.I.12 Download of an e-AD (UC2.51)</i>	118
III.I.12.1 Download of an e-AD successful	118
III.I.12.2 Download of an e-AD failed.....	121
<i>III.I.13 General query to retrieve an e-AD (UC2.52)</i>	122
SUB-SECTION III.II EXPORT SCENARIOS	124
<i>III.II.1 Local Clearance at Export</i>	125
III.II.1.1 Local Clearance at Export followed by Export Confirmation of Exit (UC2.44).....	125
III.II.1.1.1 Local Clearance at Export (UC2.44)	125
III.II.1.1.2 Export Confirmation of Exit (UC2.46).....	126
III.II.1.2 Local Clearance at Export followed by Export Cancellation of Exit.....	128
III.II.1.2.1 Local Clearance at Export (UC2.44)	128
III.II.1.2.2 Export Cancellation of Exit (UC2.46).....	128
III.II.1.2.3 Change of Destination (UC2.05).....	128
III.II.1.3 Local Clearance at Export followed by export declaration cancellation and Change of Destination	130
III.II.1.3.1 Local Clearance at Export (UC2.44)	130
III.II.1.3.2 Change of Destination (UC2.05).....	130
III.II.1.4 Local Clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration	132
III.II.1.4.1 Local Clearance at Export (UC2.44)	132
III.II.1.5 Local Clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation.....	135
III.II.1.5.1 Local Clearance at Export (UC2.44)	135

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

III.II.1.5.2 Cancellation of e-AD (UC2.10).....	135
III.II.1.6 Local Clearance at Export and movement not released by Customs followed by e-AD cancellation.....	136
III.II.1.6.1 Local Clearance at Export (UC2.44)	136
III.II.1.6.2 Cancellation of e-AD (UC2.10).....	136
III.II.1.7 Local Clearance at Export and movement not released by Customs followed by submission of new export declaration.....	137
<i>III.II.2 Export Operation at Office of Export when MSA of Dispatch is MSA of export as well (UC2.43).....</i>	<i>139</i>
III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit.....	139
III.II.2.1.1 Export Operation at office of export (UC2.43)	139
III.II.2.1.2 Export confirmation of Exit (UC2.46)	140
III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit.....	142
III.II.2.2.1 Export Operation at office of export (UC2.43)	142
III.II.2.2.2 Export Cancellation of Exit (UC2.46).....	142
III.II.2.2.3 Change of Destination (UC2.05).....	143
III.II.2.3 Export Operation at Office of Export when MSA of Dispatch is MSA of export as well, followed by export declaration cancellation and Change of Destination.....	144
III.II.2.3.1 Export Operation at office of export (UC2.43)	144
III.II.2.3.2 Change of Destination (UC2.05).....	144
III.II.2.4 Export Operation at Office of Export followed by negative cross-checking <i>before</i> the export release and Change of Destination.....	146
III.II.2.4.1 Export Operation at office of export (UC2.43)	146
III.II.2.4.2 Change of Destination (UC2.05).....	147
III.II.2.5 Export Operation at Office of Export followed by negative cross-checking <i>after</i> the export release, export declaration cancellation and submission of new export declaration	148
III.II.2.5.1 Export Operation at office of export (UC2.43)	148
III.II.2.6 Export Operation at Office of Export followed by negative cross-checking <i>after</i> the export release, export declaration cancellation and submission of change of destination	151
III.II.2.6.1 Export Operation at office of export (UC2.43)	151
III.II.2.6.2 Change of Destination (UC2.05).....	152
III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration	153
III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination	155
<i>III.II.3 Export Operation at Office of Export when MSA of Dispatch is different from MSA of export (UC2.43).....</i>	<i>156</i>
III.II.3.1 Export Operation at Office of Export followed by Export Confirmation of Exit (UC2.43).....	156
III.II.3.1.1 Export Operation at office of export (UC2.43)	157
III.II.3.1.2 Export confirmation of Exit (UC2.46)	157
III.II.3.2 Export Operation at Office of Export followed by Export Cancellation of exit.....	159
III.II.3.2.1 Export Operation at office of export (UC2.43)	159

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

III.II.3.2.2 Export Cancellation of Exit (UC2.46).....	159
III.II.3.3 Export Operation at Office of Export followed by cross-checking failure <i>before</i> the export release and Change of Destination.....	161
III.II.3.3.1 Export Operation at office of export (UC2.43)	161
III.II.3.3.2 Change of Destination (UC2.05).....	162
III.II.3.4 Export Operation at Office of Export when MSA of Dispatch is different from MSA of export, followed by export declaration cancellation and Change of Destination	163
III.II.3.4.1 Export Operation at office of export (UC2.43)	163
III.II.3.4.2 Change of Destination (UC2.05).....	164
III.II.3.5 Export Operation at Office of Export followed by cross-checking failure <i>after</i> the export release, export declaration cancellation and submission of new export declaration	165
III.II.3.5.1 Export Operation at office of export (UC2.43)	165
III.II.3.6 Export Operation at Office of Export followed by cross-checking failure <i>after</i> the export release, export declaration cancellation and submission of change of destination	168
III.II.3.6.1 Export Operation at office of export (UC2.43)	168
III.II.3.6.2 Change of Destination (UC2.05).....	168
III.II.3.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration	170
III.II.3.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination	172
SUB-SECTION III.III STATE-TRANSITION DIAGRAMS FOR CENTRAL CIRCUIT SCENARIOS ..	174
III.III.1 STD at Dispatch	174
III.III.2 STD at Destination	177
SUB-SECTION III.IV STATE-TRANSITION DIAGRAMS FOR EXPORT SCENARIOS	179
III.IV.1 Local Clearance at Export	179
III.IV.1.1 STD at Dispatch	179
III.IV.1.2 STD at Destination.....	180
III.IV.2 Export operation at Office of Export.....	181
III.IV.2.1 STD at Dispatch	181
III.IV.2.1.1 When MS of Dispatch is MS of Export as well.....	181
III.IV.2.1.2 When MS of Dispatch is different than MS of Export	182
III.IV.2.2 STD at Destination.....	183
III.IV.2.2.1 When MS of Dispatch is MS of export as well.....	183
III.IV.2.2.2 When MS of Dispatch is different than MS of Export	183
SUB-SECTION III.V FUNCTIONAL TIMERS	184
SUB-SECTION III.VI EXCEPTION HANDLING	185
III.VI.1 Rejection due to functional errors.....	185
III.VI.2 Technical Mechanism of IE904/IE905	186
III.VI.2.1 Generic scenarios	187
III.VI.2.1.1 Manual Status Request/Response	187
III.VI.2.1.1.1 Manual Status Request/Response for a known ARC.....	188
III.VI.2.1.1.2 Manual Status Request/Response for an unknown ARC.....	190
III.VI.2.1.2 Manual Status Synchronisation Request.....	192
III.VI.2.1.2.1 Manual Status Synchronisation Request for a known ARC	194
III.VI.2.1.2.2 Manual Status Synchronisation Request for an unknown ARC ..	196
III.VI.2.2 Manual Status Request/Response	198

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

III.VI.2.3 Manual Status Synchronisation Request.....	201
III.VI.2.3.1 Submission of RoR and the e-AD is under the ‘Accepted’ state at the MSA of Dispatch.....	201
III.VI.2.3.2 Change of MSA of Destination and the e-AD is under the ‘Accepted’, or ‘Partially Refused’ or ‘Refused’ state at the MSA of Destination.....	204
III.VI.2.3.3 Cancellation and the e-AD is under the ‘Accepted’ state at the MSA of Destination	207
III.VI.2.3.4 e-AD Manual Closure and the e-AD is under the ‘Accepted’ state at the MSA of Destination	210
III.VI.2.3.5 Submission of e-AD rejection and the e-AD is under the ‘Accepted’ state at the MSA of Dispatch.....	213
III.VI.2.4 Automatic Status Synchronisation Request.....	216
III.VI.2.4.1 Receipt of RoR for Unknown ARC	216
III.VI.2.4.1.1 Missed e-AD	217
III.VI.2.4.1.2 Unknown e-AD	219
III.VI.2.4.2 TIM_EAD timer expiration after Missed RoR	221
III.VI.3 Manual Closing of the Movement.....	223
SECTION IV FOLLOW-UP AND COLLABORATION	227
SUB-SECTION IV.I FOLLOW-UP SCENARIOS.....	227
IV.I.1 Submission of an event report (UC3.24).....	227
IV.I.1.1 Event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination	228
IV.I.1.2 Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch.....	231
IV.I.1.3 Event report submitted by the MSA of Destination and event occurred at the MSA of Destination	233
IV.I.1.4 Event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination	236
IV.I.1.5 Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination	239
IV.I.1.6 Event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch.....	242
IV.I.1.7 Event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination	244
IV.I.2 Control and submission of the control report (UC3.03).....	247
IV.I.2.1 Control report submitted by the MSA of Dispatch.....	248
IV.I.2.2 Control report submitted by the MSA of Destination	249
IV.I.2.3 Control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination	252
IV.I.3 Interruption of a movement (UC3.05).....	255
IV.I.3.1 Interruption of a movement submitted by the MSA of Dispatch	256
IV.I.3.2 Interruption of a movement submitted by the MSA of Destination	258
IV.I.3.3 Interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination.....	260
SUB-SECTION IV.II ADMINISTRATIVE COOPERATION SCENARIOS	263
IV.II.1 Spontaneous information (UC3.01).....	263
IV.II.2 Request for assistance.....	264
IV.II.2.1 Submission of a request for assistance	265

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

IV.II.2.2 Reminder at expiry time for sending the results to a request for assistance	266
IV.II.2.3 Submission of an answer for extending the deadline to a request for assistance	267
IV.II.2.4 Submission of the results to a request for assistance	269
<i>IV.II.3 State-Transition Diagrams for Request for assistance scenarios</i>	270
IV.II.3.1 Requesting MSA	270
IV.II.3.2 Requested MSA	270
SUB-SECTION IV.III ACCESS TO THE HISTORY OF MOVEMENTS	271
<i>IV.III.1 Request for history information</i>	271
IV.III.1.1 Submission of a history information request	272
IV.III.1.2 Reminder at expiry time for sending the results to a history information request	273
IV.III.1.3 Submission of an answer for extending the deadline to a history information request	275
IV.III.1.4 Submission of an answer for refusal of a history information request	276
IV.III.1.5 Submission of the results to a history information request	277
IV.III.1.6 Submission of partial results to a history information request	279
<i>IV.III.2 State-Transition Diagrams for Request for history scenarios</i>	280
IV.III.2.1 Requesting MSA	280
IV.III.2.2 Requested MSA	281
SUB-SECTION IV.IV MOVEMENT VERIFICATION FOR DUTY PAID MOVEMENTS SCENARIOS	282
<i>IV.IV.1 Submission of a movement verification request</i>	283
<i>IV.IV.2 Reminder at expiry time for sending the results to a movement verification request</i>	284
<i>IV.IV.3 Submission of an answer for extending the deadline</i>	285
<i>IV.IV.4 Submission of the results to a movement verification request</i>	286
<i>IV.IV.5 State-Transition Diagrams for Movement verification for duty-paid movements scenarios</i>	288
IV.IV.5.1 Requesting MSA	288
IV.IV.5.2 Requested MSA	288
<i>IV.IV.6 Functional Timers</i>	289
SECTION V CENTRAL SERVICES	290
SUB-SECTION V.I SEED	290
<i>V.I.1 Messages involved</i>	290
<i>V.I.2 Exchange of Reference Data</i>	291
V.I.2.1 Introduction	291
V.I.2.1.1 Dissemination of Reference Data (UC1.06)	291
V.I.2.1.2 Re-synchronisation of Reference Data (UC1.05)	292
V.I.2.2 Functional ways to get data	293
V.I.2.2.1 Overview	293
V.I.2.2.2 Retrieval	293
V.I.2.2.3 Extraction	293
V.I.2.2.4 Acquisition	294
V.I.2.2.5 Queries	294
V.I.2.3 Modes of access to SEED	294
V.I.2.3.1 CCN/CSI Queue Based Mode	294
V.I.2.3.2 Web Services (HTTP/S)	295
V.I.2.3.2.1 Re-synchronisation of Reference Data	295

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

V.I.2.3.3 Interactive mode.....	298
V.I.3 Exchange of SEED data	298
V.I.3.1 Introduction.....	298
V.I.3.1.1 The Role of SEED data.....	298
V.I.3.1.2 Dissemination of SEED data (UC1.14)	300
V.I.3.1.3 Re-synchronisation of SEED data (UC1.16)	302
V.I.3.2 Functional ways to get data.....	303
V.I.3.2.1 Overview	303
V.I.3.2.2 Retrieval	303
V.I.3.2.3 Extraction	304
V.I.3.2.4 Acquisition	304
V.I.3.2.5 Queries	304
V.I.3.3 Modes of access to SEED	304
V.I.3.3.1 CCN/CSI Queue Based Mode.....	304
V.I.3.3.2 Web Services (HTTP/S).....	305
V.I.3.3.2.1 Dissemination of SEED data (Reception of Updates from the MSAs)	306
V.I.3.3.2.2 Re-synchronisation of SEED data.....	310
V.I.4 SEED statistics	313
V.I.4.1 The Role of SEED statistics.....	313
V.I.4.2 SEED statistics (UC3.16).....	314
SUB-SECTION V.II CS/MISE	314
V.II.1 Messages involved.....	315
V.II.2 Manage unavailability (UC0.07)	315
V.II.2.1 Message-Based Registration of Unavailability	316
V.II.2.2 Dissemination of Unavailability	318
V.II.2.3 Exceptional Scenarios.....	318
V.II.2.3.1 Syntactically Invalid Unavailability	319
V.II.2.3.2 Semantically Invalid Unavailability	319
V.II.2.4 Unavailability Semantic Validations	320
SECTION VI SYSTEM ADMINISTRATION.....	323
SECTION VII TECHNICAL MESSAGE STRUCTURE	324
V.II.1 Introduction	324
V.II.2 Data dictionary.....	324
V.II.2.1 Data Items.....	324
V.II.2.2 Data Groups.....	324
V.II.2.3 Codelists	324
V.II.3 Technical message structure.....	325
V.II.4 Common message header	327
V.II.5 DDNEA consistency.....	328
SECTION VIII DESIGN PRINCIPLES.....	329
VIII.1.1 Approach	329
VIII.1.2 Character Sets and Data Item Conventions	329
VIII.1.2.1 Common Domain exchanges	330
VIII.1.2.1.1 Data Item conventions	330
VIII.1.2.1.1.1 Numerical Fields	330

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

VIII.I.2.1.1.2 Date/Time Fields.....	332
VIII.I.2.1.1.3 Binary data	333
VIII.I.2.1.1.4 Language Indicator for Language-sensitive text fields.....	333
VIII.I.2.1.2 Character set usage.....	333
VIII.I.2.2 External Domain exchanges.....	333
<i>VIII.I.3 Exception Handling</i>	<i>333</i>
VIII.I.3.1 Introduction	333
VIII.I.3.2 Exception hierarchy	334
VIII.I.3.2.1 Transport layer	335
VIII.I.3.2.2 Syntactic layer.....	336
VIII.I.3.2.3 Semantic layer.....	338
VIII.I.3.2.3.1 Coordination protocol validations.....	340
VIII.I.3.2.3.2 Business compliance validations	342
<i>VIII.I.4 Availability and Performance Constraints.....</i>	<i>344</i>
SECTION IX XML FORMATTING	346
<i>IX.I.1 XML Schema</i>	<i>346</i>
IX.I.1.1 XML namespaces	346
<i>IX.I.2 Character set support.....</i>	<i>347</i>
<i>IX.I.3 XML mapping of Information Exchange.....</i>	<i>348</i>
<i>IX.I.4 XML schema mapping of Information Exchange (XSD).....</i>	<i>348</i>
SECTION X TRANSPORT OF MESSAGES VIA CCN/CSI	349
<i>X.I.1 Introduction</i>	<i>349</i>
X.I.1.1 Summary	349
X.I.1.2 Architectural Assumptions.....	349
<i>X.I.2 Topology.....</i>	<i>350</i>
X.I.2.1 References to CCN/CSI	350
X.I.2.2 The CCN communication reminder	351
X.I.2.2.1 The message descriptor	352
X.I.2.2.2 The data descriptor.....	356
X.I.2.2.3 Allocation of a CSIDD.....	356
X.I.2.2.4 Inserting the application data into the CSIDD structure	357
X.I.2.2.5 Encoding the CSIDD for EMCS	357
X.I.2.2.6 The quality of service.....	360
X.I.2.2.7 Illustration of the use of the QOS parameters.....	363
X.I.2.2.8 Connecting the application to the CCN Gateway	366
X.I.2.2.9 Creating a security context for an application	366
X.I.2.2.10 Connecting to the queue manager	367
X.I.2.2.11 Opening a queue.....	367
X.I.2.2.12 CSI verbs allowed for queue accesses	369
X.I.2.2.12.1 Putting a message into a queue: HL_mq_put().....	369
X.I.2.2.12.2 Putting a message into a queue: HL_mq_put1().....	370
X.I.2.2.12.3 Reading a message from a queue: HL_mq_get().....	370
X.I.2.2.12.4 Browsing through a queue: HL_mq_browse().....	372
X.I.2.2.12.5 Deleting an element from a queue: HL_mq_delete().....	373
<i>X.I.3 Environment</i>	<i>373</i>
X.I.3.1 National Gateways	376
X.I.3.1.1 Queue Name.....	376

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

X.I.3.1.2 Gateway Site Names	377
X.I.3.2 Taxation and Customs Union DG Gateways (DG TAXUD).....	378
X.I.3.2.1 Queue Name.....	378
X.I.3.2.2 Gateway Names	379
X.I.3.3 Queue usage Overview	379
X.I.3.3.1 Operational Environment.....	380
X.I.3.3.2 Common Domain Testing Environment.....	381
X.I.4 <i>Recommended Use of CCN/CSI</i>	381
X.I.4.1 Main routines	381
X.I.4.2 Program connection	383
X.I.5 <i>Configuration Information</i>	384
X.I.5.1 Introduction.....	384
X.I.5.2 Configuration information to be provided by the MSA.....	385
X.I.5.2.1 Collection of External Configuration Data.....	385
X.I.5.2.2 Message configuration procedure	386
X.I.5.3 Configuration information to be provided by the EMCS-CO.....	387
X.I.5.3.1 Collection of External Configuration Data	387
X.I.5.3.1.1 ccnDefaultQOS	387
X.I.5.3.1.2 ccnGatewayName	388
X.I.5.3.1.3 ccnOrganisationName.....	388
X.I.5.3.1.4 ccnMessageId.....	388
X.I.5.3.1.5 ccnMessageFormalDefinition	389
X.I.5.3.1.6 ccnUserProfileId	389
X.I.5.3.2 Message configuration procedure for EMCS.....	390
SECTION XI TRANSPORT OF MESSAGES VIA SOAP/HTTP	394
SUB-SECTION XI.I TOPOLOGY	394
X.I.1.1 <i>HTTP Transport</i>	394
X.I.1.2 <i>Uniform Resource Identifier</i>	395
X.I.1.3 <i>Environments - Web Service Relative Path</i>	395
SUB-SECTION XI.II CCN CONFIGURATION	397
X.I.II.1 <i>Responsibility Model</i>	397
X.I.II.2 <i>CCN/HTTP Configuration Data</i>	397
X.I.II.3 <i>CCN/HTTP Configuration Procedure</i>	399
SUB-SECTION XI.III WEB SERVICE STANDARDS.....	400
X.I.III.1 <i>HTTP</i>	400
X.I.III.2 <i>SOAP</i>	400
X.I.III.2.1 <i>Encoding Style</i>	401
X.I.III.3 <i>WSDL</i>	401
X.I.III.4 <i>UDDI</i>	401
X.I.III.5 <i>WS-Security</i>	401
SUB-SECTION XI.IV RECOMMENDED USAGE.....	401
X.I.IV.1 <i>SOAP Message Structure</i>	402
X.I.IV.1.1 <i>SOAP 1.2 Envelope</i>	402
X.I.IV.1.2 <i>SOAP Body</i>	403
X.I.IV.2 <i>SOAP Message Exchange Patterns</i>	404
X.I.IV.2.1 <i>Asynchronous Message Exchange</i>	404
X.I.IV.2.1.1 <i>Conversation Lifetime</i>	406
X.I.IV.2.2 <i>Exception Handling</i>	407

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Table of Contents	

XI.IV.2.2.1 System Exceptions	407
XI.IV.2.2.2 Service Exceptions.....	408
XI.IV.2.2.3 Functional Errors	409
SUB-SECTION XI.V WEB SERVICE TRANSACTIONS.....	410
<i>XI.V.1 Asynchronous Web Services.....</i>	<i>410</i>
SUB-SECTION XI.VI WEB SERVICE SECURITY	411
<i>XI.VI.1 HTTP Transport Security</i>	<i>411</i>
XI.VI.1.1 CCN Authentication and Authorisation.....	411
SUB-SECTION XI.VII CCN USER HTTP AUTHENTICATION	412
SECTION XII APPLICATION AUTHENTICATION INTRANET SERVICES	414
SUB-SECTION XII.I CCNSERVERLOGIN SERVICE	414
<i>XII.I.1 DTD of the response</i>	<i>414</i>
<i>XII.I.2 Example of request.....</i>	<i>414</i>
<i>XII.I.3 Example of successful response.....</i>	<i>415</i>
<i>XII.I.4 Example of unsuccessful response.....</i>	<i>415</i>
SUB-SECTION XII.II CCNSERVERLOGOUT SERVICE.....	415
<i>XII.II.1 DTD of the response.....</i>	<i>416</i>
<i>XII.II.2 Example of request</i>	<i>416</i>
<i>XII.II.3 Example of successful response.....</i>	<i>416</i>
<i>XII.II.4 Example of unsuccessful response.....</i>	<i>416</i>
<i>XII.II.5 WS-Security</i>	<i>417</i>
APPENDICES	419

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

LIST OF FIGURES

FIGURE 1: TIME SEQUENCE DIAGRAM.....	63
FIGURE 2: EXAMPLE OF STATE TRANSITION DIAGRAM	64
FIGURE 3: INFORMATION EXCHANGE MAP OF EMCS PHASE 3	67
FIGURE 4: TSD - ORIGIN IS A TAX WAREHOUSE & DESTINATION IS KNOWN	70
FIGURE 5: CLD - ORIGIN IS A TAX WAREHOUSE & DESTINATION IS KNOWN.....	70
FIGURE 6: TSD - ORIGIN IS A TAX WAREHOUSE & DESTINATION IS UNKNOWN.....	71
FIGURE 7: CLD - ORIGIN IS A TAX WAREHOUSE & DESTINATION IS UNKNOWN	72
FIGURE 8: TSD - ORIGIN IS IMPORT AND THE E-AD IS INCONSISTENT WITH THE IMPORT DATA .	73
FIGURE 9: CLD -ORIGIN IS IMPORT AND THE E-AD IS INCONSISTENT WITH THE IMPORT DATA .	73
FIGURE 10: TSD - ORIGIN IS IMPORT AND THE DESTINATION IS KNOWN.....	75
FIGURE 11: CLD - ORIGIN IS IMPORT AND THE DESTINATION IS KNOWN	75
FIGURE 12: TSD - ORIGIN IS IMPORT AND THE DESTINATION IS UNKNOWN	77
FIGURE 13: CLD - ORIGIN IS IMPORT AND THE DESTINATION IS UNKNOWN.....	77
FIGURE 14: TSD - E-AD ALERTED.....	79
FIGURE 15: CLD - E-AD ALERTED	79
FIGURE 16: TSD - E-AD REJECTED	80
FIGURE 17: CLD - E-AD REJECTED	81
FIGURE 18: TSD - CANCELLATION OF AN E-AD BY THE CONSIGNOR (UC2.10).....	83
FIGURE 19: CLD - CANCELLATION OF AN E-AD BY THE CONSIGNOR (UC2.10)	83
FIGURE 20: TSD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “ACCEPTED” (WITH OR WITHOUT SHORTAGES)	85
FIGURE 21: CLD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “ACCEPTED” (WITH OR WITHOUT SHORTAGES)	86
FIGURE 22: TSD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “REFUSED”	87
FIGURE 23: CLD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “REFUSED”	87
FIGURE 24: TSD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “PARTIALLY REFUSED”	88
FIGURE 25: CLD - SUBMISSION OF E-AD OF WHICH DELIVERY IS “PARTIALLY REFUSED”	89
FIGURE 26: TSD - CHANGE OF MS OF DESTINATION	92
FIGURE 27: CLD - CHANGE OF MS OF DESTINATION.....	92
FIGURE 28: TSD - CHANGE MS OF DESTINATION WHERE THE CONSIGNOR DIVERTS A MOVEMENT BACK TO A (FORMER) MSA OF DESTINATION.....	95
FIGURE 29: CLD - CHANGE MS OF DESTINATION WHERE THE CONSIGNOR DIVERTS A MOVEMENT BACK TO A (FORMER) MSA OF DESTINATION.....	96
FIGURE 30: TSD - RETURN TO THE TAX WAREHOUSE OF DISPATCH.....	97
FIGURE 31: CLD - RETURN TO THE TAX WAREHOUSE OF DISPATCH	98
FIGURE 32: TSD - CHANGE OF CONSIGNEE FOLLOWING THE SUBMISSION OF E-AD (MS OF DESTINATION UNCHANGED)	100
FIGURE 33: CLD - CHANGE OF CONSIGNEE FOLLOWING THE SUBMISSION OF E-AD (MS OF DESTINATION UNCHANGED)	100
FIGURE 34: TSD - CHANGE OF PLACE OF DELIVERY FOLLOWING THE SUBMISSION OF E-AD (CONSIGNEE UNCHANGED).....	101
FIGURE 35: CLD - CHANGE OF PLACE OF DELIVERY FOLLOWING THE SUBMISSION OF E-AD (CONSIGNEE UNCHANGED).....	102
FIGURE 36: TSD - SPLITTING OF CONSIGNMENT (UC2.36).....	105
FIGURE 37: CLD - SPLITTING OF CONSIGNMENT (UC2.36).....	106
FIGURE 38: TSD - STATUS REQUEST/RESPONSE AFTER THE TIM_EAD TIMER EXPIRATION - REMINDER FOR RoR	108

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 39: CLD - STATUS REQUEST/RESPONSE AFTER THE TIM_EAD TIMER EXPIRATION - REMINDER FOR RoR	109
FIGURE 40: TSD - SUBMISSION OF EXPLANATIONS ON DELAY FOR DELIVERY - REMINDER FOR RoR	110
FIGURE 41: CLD - SUBMISSION OF EXPLANATIONS ON DELAY FOR DELIVERY - REMINDER FOR RoR	110
FIGURE 42: TSD - SENDING REMINDER MESSAGE AFTER EXPIRY OF THE CHANGE OF DESTINATION TIMER	111
FIGURE 43: CLD - SENDING REMINDER MESSAGE AFTER EXPIRY OF THE CHANGE OF DESTINATION TIMER	111
FIGURE 44: TSD - SENDING REMINDER MESSAGE AFTER EXPIRATION OF THE TIME LIMIT TO UPDATE THE DESTINATION FIELDS	112
FIGURE 45: CLD - SENDING REMINDER MESSAGE AFTER EXPIRATION OF THE TIME LIMIT TO UPDATE THE DESTINATION FIELDS	113
FIGURE 46: TSD - SENDING REMINDER MESSAGE AFTER EXPIRATION OF THE TIME LIMIT TO UPDATE THE DESTINATION FIELDS WITH CONSIGNOR EXPLANATIONS	113
FIGURE 47: CLD - SENDING REMINDER MESSAGE AFTER EXPIRATION OF THE TIME LIMIT TO UPDATE THE DESTINATION FIELDS WITH CONSIGNOR EXPLANATIONS	114
FIGURE 48: TSD - COMPLEMENTARY EXPLANATIONS SUBMISSION	115
FIGURE 49: CLD - COMPLEMENTARY EXPLANATIONS SUBMISSION	115
FIGURE 50: TSD - DUTIES CLAIMED BY THE MSA OF DESTINATION	117
FIGURE 51: CLD - DUTIES CLAIMED BY THE MSA OF DESTINATION	117
FIGURE 52: TSD - DUTIES CLAIMED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	118
FIGURE 53: CLD - DUTIES CLAIMED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	118
FIGURE 54: TSD - E-AD INFORMATION DOWNLOADED SUCCESSFULLY	120
FIGURE 55: CLD - E-AD INFORMATION DOWNLOADED SUCCESSFULLY	121
FIGURE 56: TSD - DOWNLOAD OF AN E-AD FAILED	122
FIGURE 57: CLD - DOWNLOAD OF AN E-AD FAILED	122
FIGURE 58: TSD - SUCCESSFUL RETRIEVAL OF E-AD(S)	123
FIGURE 59: CLD - SUCCESSFUL RETRIEVAL OF E-AD(S)	123
FIGURE 60: TSD - NO MOVEMENT FOUND OR LIMIT EXCEEDED	124
FIGURE 61: CLD - NO MOVEMENT FOUND OR LIMIT EXCEEDED	124
FIGURE 62: TSD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT	127
FIGURE 63: CLD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT	127
FIGURE 64: TSD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT	129
FIGURE 65: CLD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT	130
FIGURE 66: TSD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION	131
FIGURE 67: CLD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION	132
FIGURE 68: TSD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF NEW EXPORT DECLARATION	134

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 69: CLD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF NEW EXPORT DECLARATION 134

FIGURE 70: TSD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AND E-AD AND EXPORT DECLARATION CANCELLATION 135

FIGURE 71: CLD - LOCAL CLEARANCE AT EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AND E-AD AND EXPORT DECLARATION CANCELLATION 136

FIGURE 72: TSD - LOCAL CLEARANCE AT EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY E-AD CANCELLATION 137

FIGURE 73: CLD - LOCAL CLEARANCE AT EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY E-AD CANCELLATION 137

FIGURE 74: TSD - LOCAL CLEARANCE AT EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY SUBMISSION OF NEW EXPORT DECLARATION 138

FIGURE 75: CLD - LOCAL CLEARANCE AT EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY SUBMISSION OF NEW EXPORT DECLARATION 138

FIGURE 76: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT 141

FIGURE 77: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT 142

FIGURE 78: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT 143

FIGURE 79: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT 144

FIGURE 80: TSD - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MSA OF DISPATCH IS MSA OF EXPORT AS WELL, FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION 145

FIGURE 81: CLD - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MSA OF DISPATCH IS MSA OF EXPORT AS WELL, FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION 146

FIGURE 82: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING BEFORE THE EXPORT RELEASE AND CHANGE OF DESTINATION 147

FIGURE 83: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING BEFORE THE EXPORT RELEASE AND CHANGE OF DESTINATION 148

FIGURE 84: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF NEW EXPORT DECLARATION 150

FIGURE 85: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF NEW EXPORT DECLARATION 151

FIGURE 86: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF CHANGE OF DESTINATION 152

FIGURE 87: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY NEGATIVE CROSS-CHECKING AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF CHANGE OF DESTINATION 153

FIGURE 88: TSD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY NEW EXPORT DECLARATION 154

FIGURE 89: CLD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY NEW EXPORT DECLARATION 154

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 90: TSD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY CHANGE OF DESTINATION 155

FIGURE 91: CLD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY CHANGE OF DESTINATION 155

FIGURE 92: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT 158

FIGURE 93: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CONFIRMATION OF EXIT 159

FIGURE 94: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT 160

FIGURE 95: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY EXPORT CANCELLATION OF EXIT 161

FIGURE 96: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE BEFORE THE EXPORT RELEASE AND CHANGE OF DESTINATION 162

FIGURE 97: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE BEFORE THE EXPORT RELEASE AND CHANGE OF DESTINATION 163

FIGURE 98: TSD - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MSA OF DISPATCH IS DIFFERENT FROM MSA OF EXPORT, FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION 164

FIGURE 99: CLD - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MSA OF DISPATCH IS DIFFERENT FROM MSA OF EXPORT, FOLLOWED BY EXPORT DECLARATION CANCELLATION AND CHANGE OF DESTINATION 165

FIGURE 100: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE AFTER THE EXPORT RELEASE FOLLOWED BY EXPORT DECLARATION CANCELLATION AND RESUBMISSION OF NEW EXPORT DECLARATION 167

FIGURE 101: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE AFTER THE EXPORT RELEASE FOLLOWED BY EXPORT DECLARATION CANCELLATION AND RESUBMISSION OF NEW EXPORT DECLARATION 168

FIGURE 102: TSD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF CHANGE OF DESTINATION 169

FIGURE 103: CLD - EXPORT OPERATION AT OFFICE OF EXPORT FOLLOWED BY CROSS-CHECKING FAILURE AFTER THE EXPORT RELEASE, EXPORT DECLARATION CANCELLATION AND SUBMISSION OF CHANGE OF DESTINATION 170

FIGURE 104: TSD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY NEW EXPORT DECLARATION 171

FIGURE 105: CLD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY NEW EXPORT DECLARATION 172

FIGURE 106: TSD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY CHANGE OF DESTINATION 173

FIGURE 107: CLD - EXPORT OPERATION AT OFFICE OF EXPORT AND MOVEMENT NOT RELEASED BY CUSTOMS FOLLOWED BY CHANGE OF DESTINATION 173

FIGURE 108: STD AT MSA OF DISPATCH FOR CENTRAL CIRCUIT SCENARIOS – PART 1 175

FIGURE 109: STD AT MSA OF DISPATCH FOR CENTRAL CIRCUIT SCENARIOS – PART 2 175

FIGURE 110: STD AT MSA OF DISPATCH FOR CENTRAL CIRCUIT SCENARIOS – PART 3 176

FIGURE 111: STD AT MSA OF DESTINATION FOR CENTRAL CIRCUIT SCENARIOS – PART 1.... 178

FIGURE 112: STD AT MSA OF DESTINATION FOR CENTRAL CIRCUIT SCENARIOS – PART 2.... 178

FIGURE 113: STD AT MSA OF DESTINATION FOR CENTRAL CIRCUIT SCENARIOS – PART 3.... 179

FIGURE 114: STD AT DISPATCH - LOCAL CLEARANCE 180

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 115: STD AT DISPATCH - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MS OF DISPATCH SAME AS MS OF EXPORT	181
FIGURE 116: STD AT DISPATCH - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MS OF DISPATCH IS DIFFERENT THAN MS OF EXPORT	182
FIGURE 117: STD AT DESTINATION - EXPORT OPERATION AT OFFICE OF EXPORT WHEN MS OF DISPATCH IS DIFFERENT THAN MS OF EXPORT	183
FIGURE 118: TSD - MANUAL STATUS REQUEST/RESPONSE FOR A KNOWN ARC	189
FIGURE 119: CLD - MANUAL STATUS REQUEST/RESPONSE FOR A KNOWN ARC	189
FIGURE 120: TSD - MANUAL STATUS REQUEST/RESPONSE FOR AN UNKNOWN ARC	191
FIGURE 121: CLD - MANUAL STATUS REQUEST/RESPONSE FOR AN UNKNOWN ARC	191
FIGURE 122: TSD - MANUAL STATUS SYNCHRONISATION REQUEST FOR A KNOWN ARC	195
FIGURE 123: CLD - MANUAL STATUS SYNCHRONISATION REQUEST FOR A KNOWN ARC	196
FIGURE 124: TSD - MANUAL STATUS SYNCHRONISATION REQUEST FOR AN UNKNOWN ARC	197
FIGURE 125: CLD - MANUAL STATUS SYNCHRONISATION REQUEST FOR AN UNKNOWN ARC	198
FIGURE 126: TSD - MANUAL STATUS REQUEST/RESPONSE	200
FIGURE 127: CLD - MANUAL STATUS REQUEST/RESPONSE	200
FIGURE 128: TSD - STATUS SYNCHRONISATION REQUEST FOR A MISSING RoR (IE818) - RE-SUBMISSION OF RoR (IE818)	203
FIGURE 129: CLD - STATUS SYNCHRONISATION REQUEST FOR A MISSING RoR (IE818) - RE-SUBMISSION OF RoR (IE818)	204
FIGURE 130: TSD - STATUS SYNCHRONISATION REQUEST FOR A MISSING UPDATE MESSAGE (IE813) - RE-SUBMISSION OF UPDATE MESSAGE (IE813)	206
FIGURE 131: CLD - STATUS SYNCHRONISATION REQUEST FOR A MISSING UPDATE MESSAGE (IE813) - RE-SUBMISSION OF UPDATE MESSAGE (IE813)	207
FIGURE 132: TSD - STATUS SYNCHRONISATION REQUEST FOR A MISSING CANCELLATION NOTIFICATION (IE810) - RE-SUBMISSION OF CANCELLATION NOTIFICATION (IE810)	209
FIGURE 133: CLD - STATUS SYNCHRONISATION REQUEST FOR A MISSING CANCELLATION NOTIFICATION (IE810) - RE-SUBMISSION OF CANCELLATION NOTIFICATION (IE810)	210
FIGURE 134: TSD - STATUS SYNCHRONISATION REQUEST FOR A MISSING MANUAL CLOSURE NOTIFICATION (IE905)	212
FIGURE 135: CLD - STATUS SYNCHRONISATION REQUEST FOR A MISSING MANUAL CLOSURE NOTIFICATION (IE905)	212
FIGURE 136: TSD - STATUS SYNCHRONISATION REQUEST FOR A MISSING REJECTION NOTIFICATION (IE819) - RE-SUBMISSION OF REJECTION NOTIFICATION (IE819)	215
FIGURE 137: CLD - STATUS SYNCHRONISATION REQUEST FOR A MISSING REJECTION NOTIFICATION (IE819) - RE-SUBMISSION OF REJECTION NOTIFICATION (IE819)	216
FIGURE 138: TSD - POSITIVE RESPONSE ON A REQUESTED E-AD	218
FIGURE 139: CLD - POSITIVE RESPONSE ON A REQUESTED E-AD	219
FIGURE 140: TSD - NEGATIVE RESPONSE ON A REQUESTED E-AD	220
FIGURE 141: CLD - NEGATIVE RESPONSE ON A REQUESTED E-AD	220
FIGURE 142: TSD - STATUS REQUEST/RESPONSE AFTER THE TIM_EAD TIMER EXPIRATION - MISSED RoR	222
FIGURE 143: CLD - STATUS REQUEST/RESPONSE AFTER THE TIM_EAD TIMER EXPIRATION - MISSED RoR	223
FIGURE 144: TSD - MANUALLY CLOSING OF THE MOVEMENT AFTER THE REFUSAL OF DELIVERY	225
FIGURE 145: CLD - MANUALLY CLOSING OF THE MOVEMENT AFTER THE REFUSAL OF DELIVERY	226

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 146: TSD - EVENT REPORT SUBMITTED BY THE MSA OF EVENT THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	230
FIGURE 147: CLD - EVENT REPORT SUBMITTED BY THE MSA OF EVENT THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	231
FIGURE 148: TSD – EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT THE MSA OF DISPATCH.....	232
FIGURE 149: CLD – EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT THE MSA OF DISPATCH.....	233
FIGURE 150: TSD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND EVENT OCCURRED AT THE MSA OF DESTINATION	235
FIGURE 151: CLD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND EVENT OCCURRED AT THE MSA OF DESTINATION	236
FIGURE 152: TSD - EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	238
FIGURE 153: CLD - EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	239
FIGURE 154: TSD - EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT THE MSA OF DESTINATION	241
FIGURE 155: CLD - EVENT REPORT SUBMITTED BY THE MSA OF DISPATCH AND THE EVENT OCCURRED AT THE MSA OF DESTINATION	242
FIGURE 156: TSD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND THE EVENT OCCURRED AT THE MSA OF DISPATCH.....	243
FIGURE 157: CLD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND THE EVENT OCCURRED AT THE MSA OF DISPATCH.....	244
FIGURE 158: TSD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND THE EVENT OCCURRED AT NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	246
FIGURE 159: CLD - EVENT REPORT SUBMITTED BY THE MSA OF DESTINATION AND THE EVENT OCCURRED AT NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	247
FIGURE 160: TSD - CONTROL REPORT SUBMITTED BY THE MSA OF DISPATCH	249
FIGURE 161: CLD - CONTROL REPORT SUBMITTED BY THE MSA OF DISPATCH.....	249
FIGURE 162: TSD – CONTROL REPORT SUBMITTED BY THE MSA OF DESTINATION	251
FIGURE 163: CLD - CONTROL REPORT SUBMITTED BY THE MSA OF DESTINATION	252
FIGURE 164: TSD - CONTROL REPORT SUBMITTED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION.....	254
FIGURE 165: CLD - CONTROL REPORT SUBMITTED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION.....	255
FIGURE 166: TSD – INTERRUPTION OF A MOVEMENT SUBMITTED BY THE MSA OF DISPATCH	257
FIGURE 167: CLD - INTERRUPTION OF A MOVEMENT SUBMITTED BY THE MSA OF DISPATCH	257
FIGURE 168: TSD – INTERRUPTION OF A MOVEMENT SUBMITTED BY THE MSA OF DESTINATION	259
FIGURE 169: CLD - INTERRUPTION OF A MOVEMENT SUBMITTED BY THE MSA OF DESTINATION	260
FIGURE 170: TSD – INTERRUPTION OF A MOVEMENT SUBMITTED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	262
FIGURE 171: CLD - INTERRUPTION OF A MOVEMENT SUBMITTED BY A THIRD MSA THAT IS NEITHER THE MSA OF DISPATCH NOR THE MSA OF DESTINATION	263
FIGURE 172: TSD - SPONTANEOUS INFORMATION (UC3.01)	264
FIGURE 173: CLD - SPONTANEOUS INFORMATION (UC3.01)	264
FIGURE 174: TSD - SUBMISSION OF A REQUEST FOR ASSISTANCE	265

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 175: CLD - SUBMISSION OF A REQUEST FOR ASSISTANCE.....	266
FIGURE 176: TSD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A REQUEST FOR ASSISTANCE.....	267
FIGURE 177: CLD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A REQUEST FOR ASSISTANCE.....	267
FIGURE 178: TSD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE TO A REQUEST FOR ASSISTANCE.....	268
FIGURE 179: CLD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE TO A REQUEST FOR ASSISTANCE.....	268
FIGURE 180: TSD - SUBMISSION OF THE RESULTS TO A REQUEST FOR ASSISTANCE.....	269
FIGURE 181: CLD - SUBMISSION OF THE RESULTS TO A REQUEST FOR ASSISTANCE	270
FIGURE 182: STD AT REQUESTING MSA FOR REQUEST FOR ASSISTANCE SCENARIOS	270
FIGURE 183: STD AT REQUESTED MSA FOR REQUEST FOR ASSISTANCE SCENARIOS.....	271
FIGURE 184: TSD - SUBMISSION OF A HISTORY INFORMATION REQUEST.....	273
FIGURE 185: CLD- SUBMISSION OF A HISTORY INFORMATION REQUEST	273
FIGURE 186: TSD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A REQUEST FOR ASSISTANCE.....	274
FIGURE 187: CLD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A REQUEST FOR ASSISTANCE.....	274
FIGURE 188: TSD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE TO A HISTORY INFORMATION REQUEST.....	275
FIGURE 189: CLD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE TO A HISTORY INFORMATION REQUEST.....	276
FIGURE 190: TSD - SUBMISSION OF AN ANSWER FOR REFUSAL OF A HISTORY INFORMATION REQUEST.....	277
FIGURE 191: CLD - SUBMISSION OF AN ANSWER FOR REFUSAL OF A HISTORY INFORMATION REQUEST.....	277
FIGURE 192: TSD - SUBMISSION OF THE RESULTS TO A HISTORY INFORMATION REQUEST	278
FIGURE 193: CLD - SUBMISSION OF THE RESULTS TO A HISTORY INFORMATION REQUEST.....	278
FIGURE 194: TSD - SUBMISSION OF PARTIAL RESULTS TO A HISTORY INFORMATION REQUEST.....	280
FIGURE 195: CLD - SUBMISSION OF PARTIAL RESULTS TO A HISTORY INFORMATION REQUEST	280
FIGURE 196: STD AT REQUESTING MSA FOR REQUEST FOR HISTORY INFORMATION SCENARIOS	281
FIGURE 197: STD AT REQUESTED MSA FOR REQUEST FOR HISTORY INFORMATION SCENARIOS	282
FIGURE 198: TSD - SUBMISSION OF A MOVEMENT VERIFICATION REQUEST	283
FIGURE 199: CLD - SUBMISSION OF A MOVEMENT VERIFICATION REQUEST.....	284
FIGURE 200: TSD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A MOVEMENT VERIFICATION REQUEST.....	285
FIGURE 201: CLD - REMINDER AT EXPIRY TIME FOR SENDING THE RESULTS TO A MOVEMENT VERIFICATION REQUEST.....	285
FIGURE 202: TSD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE.....	286
FIGURE 203: CLD - SUBMISSION OF AN ANSWER FOR EXTENDING THE DEADLINE.....	286
FIGURE 204: TSD - SUBMISSION OF THE RESULTS TO A MOVEMENT VERIFICATION REQUEST..	287
FIGURE 205: CLD - SUBMISSION OF THE RESULTS TO A MOVEMENT VERIFICATION REQUEST .	287
FIGURE 206: STD AT REQUESTING MSA FOR MOVEMENT VERIFICATION FOR DUTY PAID MOVEMENTS SCENARIOS	288

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Figures	

FIGURE 207: STD AT REQUESTED MSA FOR MOVEMENT VERIFICATION FOR DUTY PAID MOVEMENTS SCENARIOS	289
FIGURE 208: DISSEMINATION OF REFERENCE DATA	292
FIGURE 209: RE-SYNCHRONISATION OF REFERENCE DATA	292
FIGURE 210: RETRIEVE OR EXTRACT ENTITY WEB SERVICE.....	295
FIGURE 211: DISSEMINATION OF SEED DATA FROM MSA CENTRAL SERVICES TO SEED	300
FIGURE 212: DISSEMINATION OF SEED DATA WHERE REFUSAL OF ECONOMIC OPERATORS UPDATE OCCURS	301
FIGURE 213: DISSEMINATION OF SEED DATA WHERE UPDATE OF ECONOMIC OPERATORS IS ACCEPTED.....	302
FIGURE 214: RE-SYNCHRONISATION OF SEED DATA	303
FIGURE 215: MAINTAIN ENTITY WEB SERVICE.....	306
FIGURE 216: MAINTAIN ENTITY WEB SERVICE (REJECTION OF UPDATES)	309
FIGURE 217: RETRIEVE OR EXTRACT ENTITY WEB SERVICE.....	310
FIGURE 218: SEED STATISTICS	314
FIGURE 219: SEQUENCE DIAGRAM FOR THE ANNOUNCEMENT OF UNAVAILABILITY	317
FIGURE 220: SEQUENCE DIAGRAM FOR THE DISSEMINATION OF UNAVAILABILITY	318
FIGURE 221: SEQUENCE DIAGRAM FOR THE ANNOUNCEMENT OF A SYNTACTICALLY INVALID UNAVAILABILITY	319
FIGURE 222: SEQUENCE DIAGRAM FOR THE ANNOUNCEMENT OF A SEMANTICALLY INVALID UNAVAILABILITY	320
FIGURE 223: CHARACTER SETS AND CONVENTIONS IN USE.....	330
FIGURE 224: EXCEPTION HIERARCHY	335
FIGURE 225: THE GENERATION OF THE CORRELID FIELD VALUE DECISION PROCESS	355
FIGURE 226: EXAMPLE OF CSIDD ALLOCATION, INITIALISATION WITH INFORMATION EXCHANGE AND ENCODING	359
FIGURE 227: NORMAL USE OF QoS PARAMETERS FOR EMCS.....	363
FIGURE 228: EXCEPTION AND EXPIRATION REPORTS	365
FIGURE 229: STATE TRANSITION DIAGRAM OF THE SENDING CSI STACK (NORMAL FLOW).....	365
FIGURE 230: NORMAL OPERATIONS WITH A NEA.....	380
FIGURE 231: NORMAL OPERATIONS WITH SEED	380
FIGURE 232: NORMAL OPERATIONS WITH CS/MISE	380
FIGURE 233: INTERNATIONAL TESTING WITH ANOTHER NEA.....	381
FIGURE 234: CONFORMANCE TESTING	381
FIGURE 235: A POSSIBLE SEQUENCE FOR USING CSI VERBS	383
FIGURE 236: HTTP OVER CCN TRANSPORT TOPOLOGY.....	394
FIGURE 237: GENERIC SOAP INTERACTION.....	400
FIGURE 238: SOAP 1.2 ENVELOPE.....	402
FIGURE 239: SOAP CONVERSATION PHASES	405
FIGURE 240: MSA RESPONSE TO SERVICE EXCEPTION.	408
FIGURE 241: HTTP FLOW (USER PART).....	412
FIGURE 242: CCN INTRANET USER LOGIN SCREEN.....	413
FIGURE 243: WSSE SECURITY STRUCTURE	417
FIGURE 244: SOAP REQUEST/RESPONSE ELEMENTS CONTAINED BY THE WSSE SECURITY STRUCTURE	418

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Tables	

LIST OF TABLES

TABLE 1: APPLICABLE DOCUMENTS.....	54
TABLE 2: APPLICABLE STANDARDS	56
TABLE 3: REFERENCE DOCUMENTS	56
TABLE 4: TERMINOLOGY.....	57
TABLE 5: ACRONYMS AND ABBREVIATIONS	60
TABLE 6: UML BUSINESS MODELLING ELEMENTS	61
TABLE 7: ROLE TYPES AND ORGANISATIONS	62
TABLE 8: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN ALERT OR REJECTION	78
TABLE 9: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN E-AD CANCELLATION.....	82
TABLE 10: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT A REPORT OF RECEIPT	84
TABLE 11: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT A CHANGE OF DESTINATION	90
TABLE 12: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT A SPLITTING OF CONSIGNMENT	103
TABLE 13: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT A REMINDER AT EXPIRY OF TIME LIMIT FOR REPORT OF RECEIPT.....	107
TABLE 14: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT COMPLEMENTARY EXPLANATIONS.....	114
TABLE 15: TIM_EAD FUNCTIONAL TIMER	184
TABLE 16: TIM_FDF FUNCTIONAL TIMER	185
TABLE 17: TIM_CHS FUNCTIONAL TIMER.....	185
TABLE 18: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SEND A REMINDER FOR ADMINISTRATIVE COOPERATION	266
TABLE 19: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN ANSWER TO A REQUEST FOR ASSISTANCE	267
TABLE 20: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT THE RESULTS TO A REQUEST FOR ASSISTANCE	269
TABLE 21: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SEND A REMINDER FOR HISTORY INFORMATION	273
TABLE 22: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN ANSWER FOR EXTENDING THE DEADLINE TO A HISTORY INFORMATION REQUEST.....	275
TABLE 23: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN ANSWER FOR REFUSING A HISTORY INFORMATION REQUEST	276
TABLE 24: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT THE RESULTS TO A HISTORY INFORMATION REQUEST.....	277
TABLE 25: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT PARTIAL RESULTS TO A HISTORY INFORMATION REQUEST	279
TABLE 26: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SEND A REMINDER FOR A MOVEMENT VERIFICATION REQUEST	284
TABLE 27: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT AN ANSWER TO A MOVEMENT VERIFICATION REQUEST	285
TABLE 28: SCENARIOS THAT MUST HAVE BEEN PREVIOUSLY COMPLETED IN ORDER TO SUBMIT THE RESULTS TO A MOVEMENT VERIFICATION REQUEST	287
TABLE 29: TIM_MVS FUNCTIONAL TIMER.....	289

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
List of Tables	

TABLE 30: USE OF STATUS CODES	326
TABLE 31: DATE/TIME FIELDS FORMAT AND THEIR CORRESPONDING REGULAR EXPRESSIONS	332
TABLE 32: SEMANTIC LAYER VALIDATIONS APPLICABILITY TO BUSINESS COMMUNICATION CHANNELS	338
TABLE 33: COORDINATION PROTOCOL ERROR CODES.....	340
TABLE 34: COORDINATION PROTOCOL ERRORS TO REFUSAL MESSAGE MAP	340
TABLE 35: BUSINESS COMPLIANCE ERROR CODES	342
TABLE 36: BUSINESS COMPLIANCE ERRORS TO REFUSAL MESSAGE MAP	342
TABLE 37: BUSINESS COMPLIANCE ERROR CODES SPECIFIC TO IE714	343
TABLE 38: BUSINESS COMPLIANCE ERROR CODES SPECIFIC TO IE702	344
TABLE 39: MQ MESSAGE DESCRIPTOR	352
TABLE 40: CSI DATA DESCRIPTOR	356
TABLE 41: MSGTypeID USED FOR AN INFORMATION EXCHANGE OF EMCS	359
TABLE 42: CCN/CSI QUALITY OF SERVICE STRUCTURE	360
TABLE 43: MQ OBJECT DESCRIPTOR	368
TABLE 44: CSI VERBS FOR QUEUE ACCESS	369
TABLE 45: CSIMQGMO OBJECT DESCRIPTOR.....	372
TABLE 46: INFORMATION EXCHANGES EXCHANGED VIA THE 'CORE FLOW' QUEUE.....	375
TABLE 47: QUEUE NAMES FOR NATIONAL GATEWAYS.....	377
TABLE 48: NATIONAL GATEWAY NAMES	378
TABLE 49: QUEUE NAMES FOR TAXATION AND CUSTOMS UNION DG GATEWAYS.....	378
TABLE 50: NATIONAL GATEWAY NAME	379
TABLE 51: EXTERNAL CONFIGURATION DATA DEFINED BY MSA.....	386
TABLE 52: EXTERNAL CONFIGURATION DATA DEFINED BY EMCS-CO.....	387
TABLE 53: CONFIGURATION OF DEFAULT QoS.....	387
TABLE 54: LIST OF USER PROFILES	390
TABLE 55: IDL DEFINITION OF CCN MESSAGES FOR EMCS	393
TABLE 56: CENTRAL SERVICES WEB SERVICE RELATIVE URIS BY PROCESSING AREA	396
TABLE 57: USER PROFILES FOR NATIONAL GATEWAYS ACCESS TO CENTRAL SERVICES WEB SERVICES	398
TABLE 58: USER PROFILES FOR NATIONAL GATEWAYS ACCESS TO CENTRAL SERVICES WEB INTERFACE	398
TABLE 59: USER PROFILES FOR DG TAXUD GATEWAY ACCESS TO CENTRAL SERVICES WEB SERVICES	398
TABLE 60: USER PROFILES FOR DG TAXUD GATEWAY ACCESS TO CENTRAL SERVICES WEB INTERFACE	398
TABLE 61: SOAP CONVERSATION LIFETIME ATTRIBUTES	406
TABLE 62: CCNSERVERLOGIN SYNTAX AND DESCRIPTION	414
TABLE 63: CCNSERVERLOGIN RESPONSE'S DTD	414
TABLE 64: CCNSERVERLOGIN REQUEST EXAMPLE	415
TABLE 65: CCNSERVERLOGIN SUCCESS RESPONSE EXAMPLE	415
TABLE 66: CCNSERVERLOGIN ERROR RESPONSE EXAMPLE	415
TABLE 67: CCNSERVERLOGOUT SYNTAX AND DESCRIPTION	415
TABLE 68: CCNSERVERLOGOUT RESPONSE'S DTD	416
TABLE 69: CCNSERVERLOGOUT REQUEST EXAMPLE	416
TABLE 70: CCNSERVERLOGOUT SUCCESS RESPONSE EXAMPLE.....	416
TABLE 71: CCNSERVERLOGOUT ERROR RESPONSE EXAMPLE	416

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

Section I GENERAL INFORMATION

This section contains some introductory information regarding the relationship of this document with the other EMCS baseline documents. Moreover, it provides some clarifications on the terminology and abbreviations used in this document as well as some information with the symbolism and conventions used.

I.I.1 Applicable and Reference documents

I.I.1.1 Applicable Documents and Standards

I.I.1.1.1 Documents

The following documents are applicable to this document:

Ref.	Reference	Title	Release
A1	ECP1-ESS-FESS	Functional Excise System Specifications (FESS)	3.32-EN
A2	ECP3-EMCSDEV-SC02-SD	Scope Document for EMCS Phase 3	1.05-EN
A3	CCN/CSI-PRG-AP/C-01-MARB	CCN/CSI Application Programming Guide (C language)	Ed11
A4	CCN/CSI-PRG-HL/Cob/BS2000-01-MARB	HL Programming Guide (COBOL Language for BS2000)	Ed01
A5	CCN/CSI-PRG-HL/CICS-01-MARB	HL Programming Guide (COBOL Language for IBM CICS environment)	Ed03
A6	CCN/CSI-REF-HL/C-01-MARB	CCN/CSI HL Reference Manual	Ed13
A7	CCN/CSI-REF-GSS/C-01-BING	CCN/CSI GSS Reference Manual	Ed05
A8	CCN/CSI-REF-ComD/C-01-MARB	CCN/CSI Common Definitions Reference Manual (C language)	Ed13
A9	CCN/CSI-REF-ERR-01-MARB	CCN/CSI Error Reason Codes Reference Manual	Ed05
A10	ECP1-ESS-SESS	Security Excise System Specifications	2.00-EN
A11	ECP1-ESS-TESS	Technical Excise System Specifications	3.00-EN
A12	ECP1-ESS-FRS	Fall-back and Recovery Specification	3.11-EN
A13	ECP1-ESS-PSS	Phasing and Scope Specification	3.30-EN
A14	TCE-L1-DDNTA_P32	DDNTA for NCTS Phase 3.2 and ECS	8.10-EN

Table 1: Applicable Documents

Note that all documents listed above are applicable to this document (and are input to this document). Any change in any of the documents above is likely to have direct and immediate consequences for this document.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

DDNEA is aligned with all applicable documents listed in Table 1.

1.1.1.1.2 Standards

The following standards are applicable to this document:

Ref.	Reference	Title
S1	XML standard 1.0 Release 3	http://www.w3.org/TR/REC-xml
S2	Unicode standard version Unicode 6.1.0	http://www.unicode.org/versions/Unicode6.1.0/
S3	UTF-8 encoding	See Section 2.5 “Encoding Forms” and Section 3.9 “ Unicode Encoding Forms ” in the Unicode Standard [S2]
S4	RFC-1630	Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web
S5	RFC-1867	Form-based File Upload in HTML
S6	RFC-1950	ZLIB Compressed Data Format Specification version 3.3
S7	RFC-1951	DEFLATE Compressed Data Format Specification version 1.3
S8	RFC-1952	GZIP file format specification version 4.3
S9	RFC-2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of message Bodies
S10	RFC-2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
S11	RFC-2047	MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text
S12	RFC-2048	Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures
S13	RFC-2049	Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples
S14	RFC-2068	Hypertext Transfer Protocol—HTTP/1.1
S15	XML Schema 1.0	http://www.w3.org/TR/xmlschema-0/
S16	SOAP 1.2	http://www.w3.org/TR/soap12-part0/
S17	SOAP 1.1	http://www.w3.org/TR/2000/NOTE-SOAP-20000508/
S18	Web Services Description Language (WSDL) 1.1	http://www.w3.org/TR/wsdl
S19	XML-binary Optimized Packaging	http://www.w3.org/TR/xop10/

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

Ref.	Reference	Title
S20	SOAP Message Transmission Optimization Mechanism	http://www.w3.org/TR/soap12-mtom/

Table 2: Applicable Standards

I.I.1.2 Reference Documents and Standards

I.I.1.2.1 Documents

The following documents are also of interest to the NEA designer:

Ref.	Reference	Title	Release
R1	ECP1-ESS-GLT	Glossary of Terms	2.04-EN
R2	ECP1-ESS-SEP	Security Policy Document	2.02-EN
R3	CCN/CSI-OVW-GEN-01-MARB	CCN/CSI System Overview	Ed09
R4	CCN/CSI-AD-GEN-01-MARB	CCN/CSI Architecture Design	Ed07
R5	CCN/CSI-TRA-CSI-01	CCN/CSI Course Notes (mod1 for Cobol)	Ed03
R6	CCN/CSI-TRA-CSI-01	CCN/CSI Course Notes (mod2 for Java)	Ed03
R7	CCN/CSI-TRA-CSI-01	CCN/CSI Course Notes (mod3 for C)	Ed01
R8	CCN/CSI-ACG-GEN-01-MARB	CCN/CSI Application Configuration Guide	Ed07
R9	CCNCSI-SIG-SRA-01	Software Installation Guide - Statistics Receiver Application	Ed00
R10	CCN-CPRG-IAS	CCN Intranet Services - Programmer's Guide	1.00-EN
R11	ECP2- FITSDEV2-SC03-SRD-SEEDv1	System Requirements Definition for SEEDv1	1.37-EN
R12	ECP3-FITSDEV2-TA-SRD	System Requirements Definition for Testing Application	1.22-EN

Table 3: Reference Documents

I.I.2 Definitions

I.I.2.1 Definitions

Definitions of many of the terms used in this document may be found in the “Glossary of Terms” [R1].

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

I.I.2.2 Terminology

A number of terms are used with a very specific meaning in this document:

Name	Description
Codelist	A set of discrete values. Some Data Items can only contain a value chosen from a set of discrete values, in which case they will have an associated Codelist.
Data Group	A Data Group is a part of the Technical Message Structure; it groups Data Items related to the same subject and it is denoted by a Data Group name.
Data Item	A Data Item is an elementary (atomic) piece of information; part of a Data Group.
Functional Message Structure (FMS)	Logical data structure of an Information Exchange, as defined in FESS.
Information Exchange	A logical exchange of information between two locations. An Information Exchange is the conceptual exchange of information between two organisations, independent of its physical means.
Location	A place where the excise business is performed.
Message formatting	Representation (of a Technical Message Structure) in or mapping to exchange syntax (e.g. XML).
Message transport	The sending (and reception) of a formatted message across a communications platform (such as CCN/CSI)
Organisation	An organisation is a number of individuals acting in a concerned way towards a common business purpose with allocated roles and responsibilities. An organisation can have one or more roles of a specific type.
Reference Data	A collection of discrete data, maintained by the SEED application and that can be sent to a NEA as an IE734. Although this term is sometimes used in order to denote <i>any</i> data maintained by SEED, DDNEA will use it in the narrow sense defined above.
Technical Message Structure (TMS)	The data structure of the Information Exchange as it needs to be implemented. A TMS is a structure (and hierarchy) of Data Groups.
Time Sequence Diagram	This diagram shows EMCS roles interactions arranged in time sequence. In particular, it shows the EMCS roles participating in the interaction and the sequence of messages exchanged.
Collaboration Diagram	This diagram describes a pattern of interaction among EMCS roles; it shows the EMCS roles participating in the interaction by their links to each other and the messages they send to each other.

Table 4: Terminology

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

II.2.3 Acronyms and Abbreviations

The following acronyms are used in this document:

Acronym	Description
AD	Administrative Document
API	Application Programming Interface
ARC	Administrative Reference Code
BCC	Business Communication Channels
CCN	Common Communication Network
CD	Common Domain
CDEA	Centrally Developed Excise Application
CDIA	CCN/TC Directory Administrator
CLD	Collaboration Diagram
CoA	Confirm on Arrival
CoD	Confirm on Delivery
CONTRL	Syntax and service report XML message
COS	Central Operation Specification
CPT	Central Project Team
CS/MISE	Central Services - Management Information System for Excise
CSI	Common Systems Interface
CSIDD	CCN/CSI Data Descriptor
DDNEA	Design Document for National Excise Applications
DG TAXUD	TAXation and Customs Union DG
DTD	Document Type Definition
EC	European Community
ECP	Excise Computerisation Project
ECS	Export Control System
ELO	Excise Liaison Office
EMCS	Excise Movement and Control System
EMCS-CO	Excise Movement and Control System - Central Operations
EXC	Exception Report
EXP	Expiration Report
FESS	Functional Excise System Specification

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

Acronym	Description
FMS	Functional Message Structure
FRS	Fallback and Recovery Specification
FS	Functional Stage
GLT	Glossary of Terms
GMT	Greenwich Mean Time
GSS	Generic Security Services
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HTTP over SSL
IE	Information Exchange
ISO	International Standards Organisation
IT	Information Technology
KEL	Known Error List
LRN	Local Reference Number
LSO	Local Security Officer
MD5	Message Digest 5
MIME	Multipurpose Internet Mail Extensions
MS	Member State
MSA	Member State Administration
NACK	Non-ACKnowledgement service message
NCTS	New Computerised Transit System
NDEA	Nationally Developed Excise Application
NEA	National Excise Application
ORO	Occasionally Registered Operator
PRO	Permanent Registered Operator
PSS	Phasing and Scope Specification
QoS	Quality of Service
RD	Reference Data
RoR	Report of Receipt
SAM	Single Administrative Message
SD	Scope Document
SEED	System for Exchange of Excise Data

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

Acronym	Description
SEP	Security Policy
SESS	Security Excise System Specification
SGML	Standard Generalised Markup Language
SMTP	Simple Mail Transfer Protocol
SRO	System Requirements Overview
SSL	Secure Socket Layer
STD	State Transition Diagram
TA	Test Application
TC	Technical Centre
TCP/IP	Transmission Control Protocol / Internet Protocol
TESS	Technical Excise System Specification
TMS	Technical Message Structure
TSD	Time Sequence Diagram
UC	Use Case
UML	Unified Modelling Language
URI	Universal Resource Identifier
UTF	UCS Transformation Format
VAT	Value Added Tax
VIIES	VAT Information Exchange System
WS	Web Service
WWW	World Wide Web
XML	Extensible Markup Language
XSD	XML Schema Definition

Table 5: Acronyms and Abbreviations

I.I.3 Symbolism and Conventions Used

This chapter presents the symbolism used in this document. It is necessary to understand this section before reading the remaining sections. An explanation of symbolism used in the appendices can be found at the beginning of the relevant Appendix.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

II.3.1 Time Sequence Diagrams

The Information Exchange sequences are presented using UML Time Sequence Diagrams (TSD) and UML Collaboration Diagrams (CLD). The Time Sequence Diagrams and Collaboration Diagrams visualise the Information Exchange sequence between all locations involved in a particular scenario for an excise movement. Each type of interaction diagram presents a different aspect of the information exchange. Finally, the involved participants in a scenario are illustrated using business-modelling elements based on the UML profile for business modelling.

The used business modelling elements are the following:

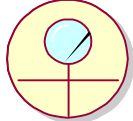
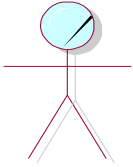
Name	Notion	Icon (Example)	UML Stereotype
Business Worker	A business worker is an abstraction of software or even a system that is a composite of these and represents a role performed within business use case realisations. A business worker collaborates with other business workers, is notified of business events and manipulates business entities to perform its responsibilities. In EMCS context, a business worker is a MSA application, which has active participation in the realisation of use cases as these are defined in FESS. Each business worker (MSA application) collaborates with other business workers (MSA applications) through the Common Domain and manipulates business entities such as data, messages in order to perform some activities (its responsibilities) as these are defined in FESS.	 : System: MSA application	<<business worker>>
Business Actor	A business actor represents a role played in relation to the business by someone or something (application) in the business environment. In EMCS, this is used for the Economic Operators, who play a significant role in the EMCS business but their applications are out of the scope of the DDNEA since the boundaries is the Common Domain. Please note that The Economic Operators (Consignor and Consignee) are presented in the scenarios in order to have an overview of the different message exchanges as well as of some exceptional cases. However, the implementation of Economic Operator's applications is not in the scope of the DDNEA since they are in the external domain.	 Economic Operator	<<business actor>>

Table 6: UML business modelling elements

Examples of scenarios for excise movements are the Core Business.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

As Time Sequence Diagrams and Collaboration Diagrams can only be used to show one possible flow of Information Exchanges, a large number of TSDs and CLDs are required to specify all allowed Information Exchange sequences.

The following **roles** that can be taken by organisations are used in this version of DDNEA:

Role type	Role name	Organisation	UML Stereotype
MSA dispatch application	Member State Administration of Dispatch	Member State Administration	<<business worker>>
MSA destination application	Member State Administration of Destination	Member State Administration	<<business worker>>
Consignor	Any Authorised Warehouse keeper	Economic Operator	<<business actor>>
Consignee	Any Economic Operator Authorised Warehouse keeper, Registered Consignee, and Temporary Registered Consignee at Destination	Economic Operator	<<business actor>>
Requesting MSA	Any Member State Administration	Member State Administration	<<business worker>>
Requested MSA	Any Member State Administration	Member State Administration	<<business worker>>
Addressed MSA application	Member State Administration of addressed application	Member State Administration	<<business worker>>
Issuing MSA application	Member State Administration of issuing application	Member State Administration	<<business worker>>
MSA event application	Member State Administration of event application	Member State Administration	<<business worker>>
Third MSA	Any Member State Administration	Member State Administration	<<business worker>>
Customs Export application	Member State of Export application	National Administration	<<business worker>>
Customs	Member State of Import application	National Administration	<<business worker>>

Table 7: Role types and organisations

Please note that the infrastructure of a MSA is not specified in DDNEA.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

For the roles of MSA of Destination and Consignee, a distinction is made in the case of Change of Destination (Section III - III.1.5) between the former MSA of Destination/Consignee and the new MSA of Destination/Consignee.

The TSDs and CLDs depict a particular scenario of message exchanges and may include one or more FESS use-cases.

The components of a Time Sequence Diagram (TSD) are shown in the following figure:

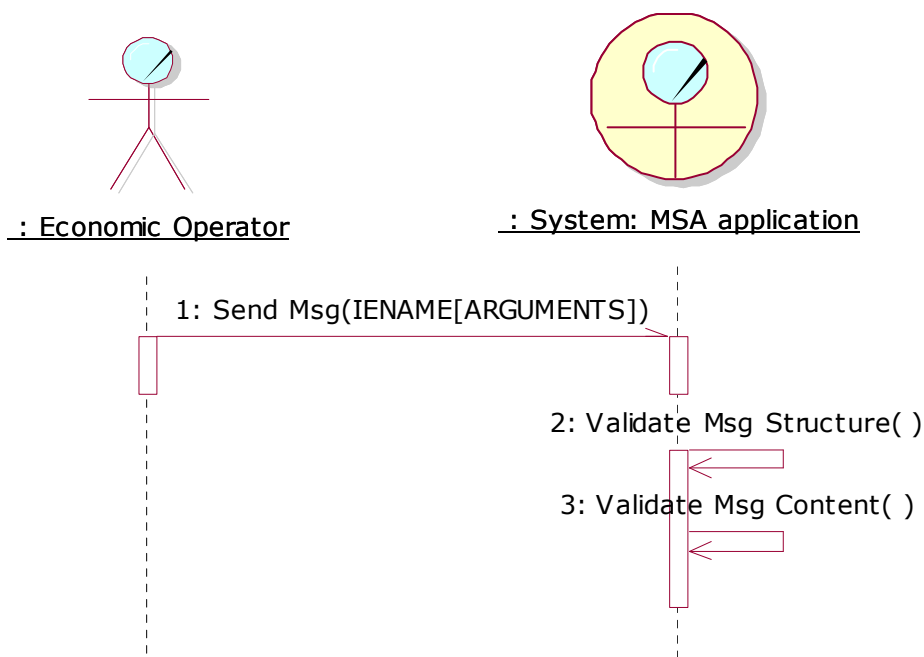


Figure 1: Time Sequence Diagram

In a TSD, each role is represented by an icon (according to Table 7) with the name of that role (Figure 1) and a vertical line, called the “lifeline”. The name “lifeline” comes from the UML-concept that an object’s life can be ended. This does not apply here.

An arrow between the lifelines represents each Information Exchange (message) between two roles, where the arrow shows the direction of the Information Exchange. Attached to the arrow, a label gives the sequence number of this Information Exchange in the scenario, the operation used to pass the message, the coded name and number of Information Exchange as well as any argument passed with the Information Exchange.

The above figure (Figure 1) illustrates that the “Economic Operator” submits a message to the MSA application using the “Send Msg” operation. The name of the IE that is sent to MSA application is indicated inside the brackets of the “Send Msg” operation. Moreover, the Square Brackets in the message above (Figure 1) indicate some arguments of the message such as in the case of IE818, which contains the acceptance of the delivery (*Send Msg (IE818: C_DEL_DAT [Acceptance])*).

The word “System:” has been added before the name of the MSA application (business worker name) to indicate that the particular business worker is a system and not a human. Moreover, the arrow with half arrowhead indicates that the message is asynchronous.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

Finally, in the State Transition Diagrams and Collaboration Diagrams of this document, two types of operations have been used indicating two different types of validations. These should be performed by the MSA applications on a received message when it is a draft and when it is duly valid (after the successful validation of the draft). In the first case, when a draft is sent by the Economic Operator, the MSA application uses the operation “Validate Msg Structure ()” to check the validity of the message against the corresponding XSD of DDNEA (this includes the structure validation as well as the validation of technical Codelists) and the operation “Validate Msg Content ()” to check the business validity of the draft (against rules, conditions, business Codelists, etc.). In the case where a message is received as a result of a successful validation of the draft from the sender MSA application, the recipient MSA application shall check only the message structure using the “Validate Msg Structure ()” operation. The exceptional cases are described in Sub-Section III.VI of Section III indicating message exchanges when at least one of the aforementioned validations is not completed successfully.

The narrow rectangles on the lifelines are called ‘focus of control’. It represents the relative time that the control of the flow is focused in that role, thus the time that the role is directing messages. When more than one message starts from (or ends in) the same focus of control, this means that these messages are sent (or received) shortly after each other. The arrows will appear close to each other in that case as well. Please note that in this case the sequence of sending the messages is not important. Therefore, the sequence used to represent them in the TSDs is only indicative. When for two messages exchanged the sequence is important they are presented to start from a different ‘focus of control’.

A similar notation is used in the Collaboration Diagrams.

1.1.3.2 State Transition Diagrams

The State Transition Diagrams (STD) consists of states and transitions between those states. Each state represents the state of an excise movement for a particular MSA role. Each transition starts at a given state and goes to another state and is triggered by the exchange of a message between two organisations. Only transitions that are triggered by the exchange of a message are shown on these State Transition Diagrams.

Every State Transition Diagram in this document is related to one particular MSA role only. For every role, it is defined how state transitions take place according to events (such as the reception of a message from another role).

States are shown as a box and an arrow shows transitions.

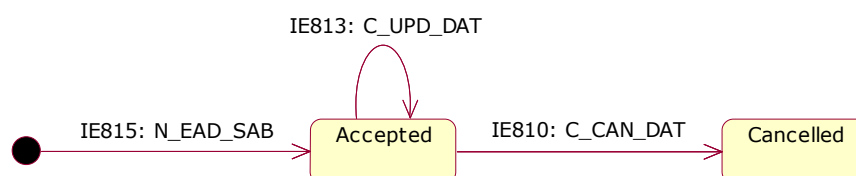


Figure 2: Example of State Transition Diagram

The State transitions show the event for these transitions. In the example above, the event for the transition from the “Accepted” state to the “Cancelled” state is the reception of cancellation message (IE810: C_CAN_DAT).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section I - General Information	

In some cases, an argument is indicated in the event as in the case of transition from “Accepted” to “Accepted” (transition to self). In this case, the state is retained to “Accepted” when the e-AD response is either positive or negative.

In EMCS, the set of names of the states in the State Transition Diagrams is a subset of those contained in FESS [A1]. A number of additional technical states have also been defined in this document. The STD for Core Business can be found in Sub-Section III.III and for Follow-up and collaboration in IV.III.1.6.

I.I.3.3 Data dictionary

The data dictionary, contained within this DDNEA, defines 3 specific items:

- Data Items;
- Data Groups;
- Codelists (sets of discrete values).

A number of naming and spelling conventions and rules have been maintained for these throughout this document. The rules are as follows:

I.I.3.3.1 Data Items

Every name shall start with a capital (uppercase) letter.

Every name can contain letters, digits, and a number of additional characters: the space character, the brackets “(” and “)”, the ampersand character “&”, the underscore character “_”, and the slash character “/”. No other characters are allowed.

Within the name, lowercase letters shall preferably be used (except for the first character and except for abbreviations that will always be in uppercase).

I.I.3.3.2 Data Groups

Every name shall start with a letter or with the bracket “(“character.

Every name can contain letters, digits, and a number of additional characters: the space character, the brackets “(” and “)”, the ampersand character “&”, the underscore character “_”, and the slash character “/”. No other characters are allowed.

Only uppercase letters are allowed.

I.I.3.3.3 Codelists

The same rules as for Data Items will apply.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER: 1.76-EN
Section II - Scope of development	

Section II SCOPE OF DEVELOPMENT

This section discusses the items that need to be developed in EMCS Phase 3 applications.

II.I.1 Information Exchange Overview for EMCS Phase 3

The scope of EMCS for Phase 3 is depicted in Appendix A. In particular:

- Appendix A1 summarises the various requirements for any NDEA and presents an overall view of the Information Exchanges to be supported in EMCS for Phase 3, as defined in PSS [A13].
- Appendix A2 performs a breakdown of the overall development for the EMCS Phase 3 in the different Business Processes (Core Business, Central Services and System Administration). Moreover, it classifies the messages into business and technical. The Appendix A2 is derived from the full message scope, which is defined in Appendix A1.

II.I.2 Information Exchange Map of EMCS Phase 3 for Core Business

The Core Business Information Exchanges to be supported in EMCS Phase 3 and the different parties involved are summarised in the diagram below (Figure 3). More detailed specifications of those message exchanges are presented in Section III. Please note that this diagram is not a Time Sequence Diagram. It only summarises the different possible sources and destinations for the various Information Exchanges. This diagram highlights in which Domain the different exchanges are to be foreseen.

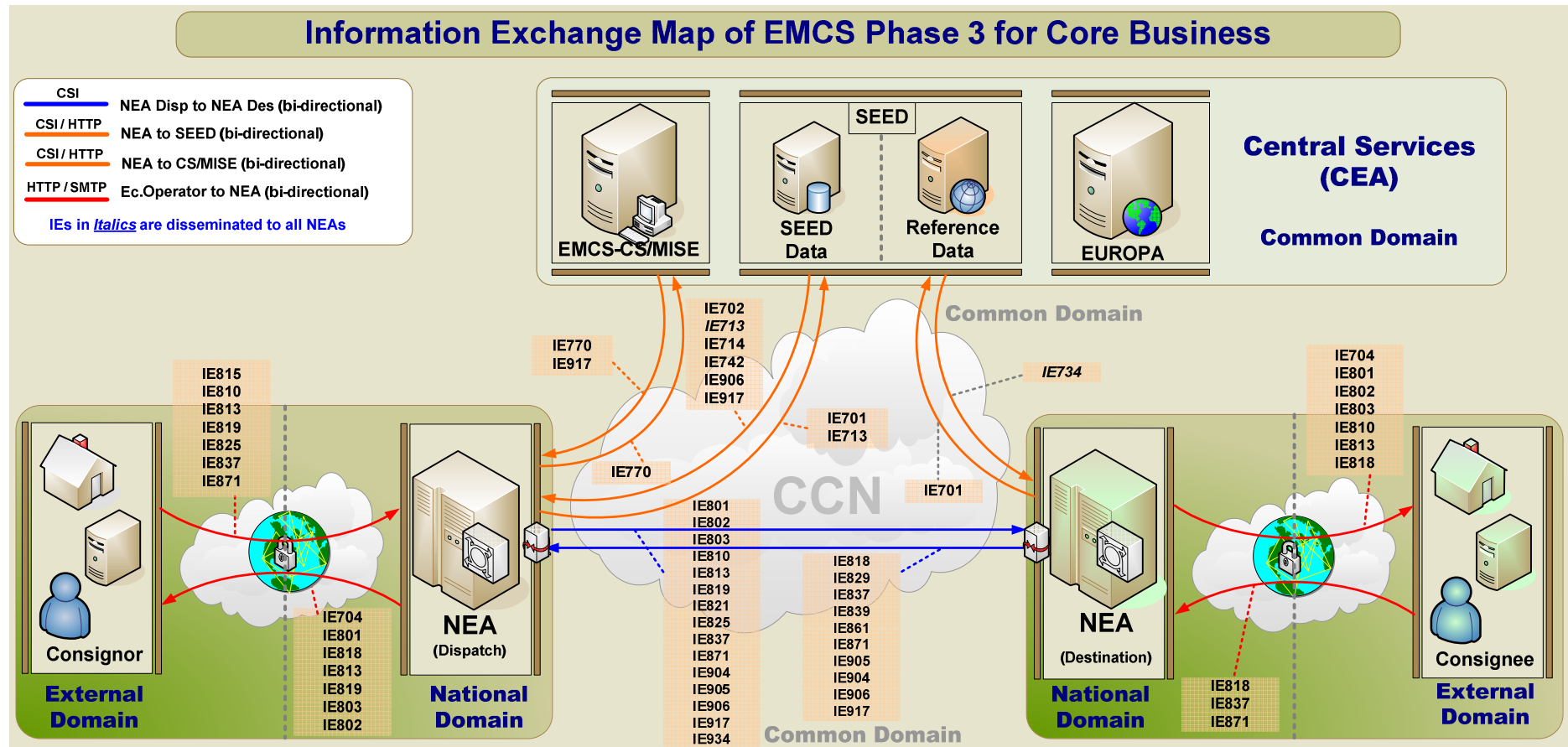


Figure 3: Information Exchange Map of EMCS Phase 3

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Section III CORE BUSINESS

The following section contains a detailed specification of the message exchange protocols to be foreseen for the EMCS Core Business area in EMCS Phase 3. These exchange protocols shall define the valid sequence of the message exchanges between the NEAs based on the Business Processes as these are defined in FESS [A1].

The different Time Sequence Diagrams should be read in conjunction with the State Transition Diagrams that have been included in Sub-Section III.III. Every application should implement both the Time Sequence Diagrams and the State Transition Diagrams logic.

Finally, it shall be noted that the following scenarios include functionality (timers, etc.), which according to the SD [A2] is optional in case the MSAs decide to implement it based on the national needs. Moreover, the scenarios below consider that the Consignee is PRO, which implies the availability of an Electronic Data Interchange (EDI) interface between the MSA destination application and the Consignee.

Sub-Section III.I Central Circuit Scenarios

This section aims to specify the most important message exchange protocols for EMCS Phase 3. The identification of the following scenarios has been based on the “Central Circuit” use cases of FESS [A1], which are in the scope of EMCS Phase 3 according to SD [A2].

Please note that two types of operations have been used in the diagrams of this chapter indicating two different types of validation that should be performed by the MSA applications on a received message when this is draft and when it is duly valid (after the successful validation of the draft). In the first case, when a draft is sent by the Economic Operator, the MSA application uses the operation “Validate Msg Structure ()” to check the validity of the message against the corresponding XSD of DDNEA (this includes the structure validation as well as the validation of technical Codelists) and the operation “Validate Msg Content ()” to check the business validity of the draft (against rules, conditions, business Codelists, etc.). In the case where a message is received as a result of a successful validation of the draft from the sender MSA application, the recipient MSA application shall check only the message structure using the “Validate Msg Structure ()” operation. The exceptional cases are described in Sub-Section III.VI indicating message exchanges when at least one of the aforementioned validations is not completed successfully.

III.I.1 Submission and registration of an e-AD (UC2.01)

This use case describes the message exchange protocol for submitting and registering an e-AD. Within the context of the EMCS, the e-AD is a message containing information about the consignment; namely the origin, the destination, the Economic Operators that are involved in the movement and the product codes along with their quantities being moved.

The different scenarios of this use case are determined by the combinations of certain types of the origin and the destination of the movement. The origin types that determine the combinations depend on whether the consignment initiates from a tax warehouse or from a place of import. The destination types that determine the combinations depend on whether the consignment has a known destination or an unknown destination. The destination type of an e-AD may be outside the community in which case the scenarios for the exportation of goods apply described in Sub-Section III.II Export Scenarios.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

For each combination of the aforementioned origin and destination types, the different scenarios are the following:

- when the origin of the movement is a tax warehouse and
 - the destination of the movement is known (see III.I.1.1.1); or
 - the destination of the movement is unknown (see III.I.1.1.2); or
- when the origin of the movement is a place of import and
 - the destination of the movement is known (see III.I.1.2.2); or
 - the destination of the movement is unknown (see III.I.1.2.3).

All scenarios begin essentially with the consignor submitting a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application that subsequently validates the structure and the message content. It should be noted that although the submission occurs before the actual dispatch of goods, if the e-AD is submitted in deferred mode, then the actual dispatch of goods would have preceded the draft e-AD submission. The exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

For the case where the origin of the movement is a place of import, an additional scenario describes the exceptional case where the e-AD has been found to be inconsistent with the import data (see III.I.1.2.1).

III.I.1.1 Origin is tax warehouse

This section describes message exchange protocol for the submission and registration of an e-AD having a tax warehouse as origin and having a tax warehouse keeper as the consignor.

The exceptional cases where the draft e-AD (IE815: N_EAD_SUB) submitted by the consignor is found to be invalid are described in Sub-Section III.VI.

III.I.1.1.1 Origin is tax warehouse and the destination is known

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The origin of the movement is a tax warehouse and the destination of the movement is known (i.e. the destination is not unknown nor it is for export).

The MSA dispatch application upon receiving the draft e-AD performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application assigns an ARC to the e-AD (the structure of ARC and the Check Digit algorithm are defined in FESS Appendix B [A1]) and creates a validated e-AD (IE801: C_EAD_VAL) with sequence number “1” that disseminates to the MSA destination application and to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “*Accepted*” and the timer TIM_EAD is initiated to expire at the expected end of movement (i.e. date of dispatch plus journey time).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the MSA destination application stores the e-AD and sets the state of the e-AD at MSA of Destination to “Accepted”. Finally, the MSA destination application forwards the e-AD (IE801: C_EAD_VAL) to the Consignee.

The scenario of the submission and registration of an e-AD having a tax warehouse as origin and a known movement destination is depicted in Figure 4 and Figure 5:

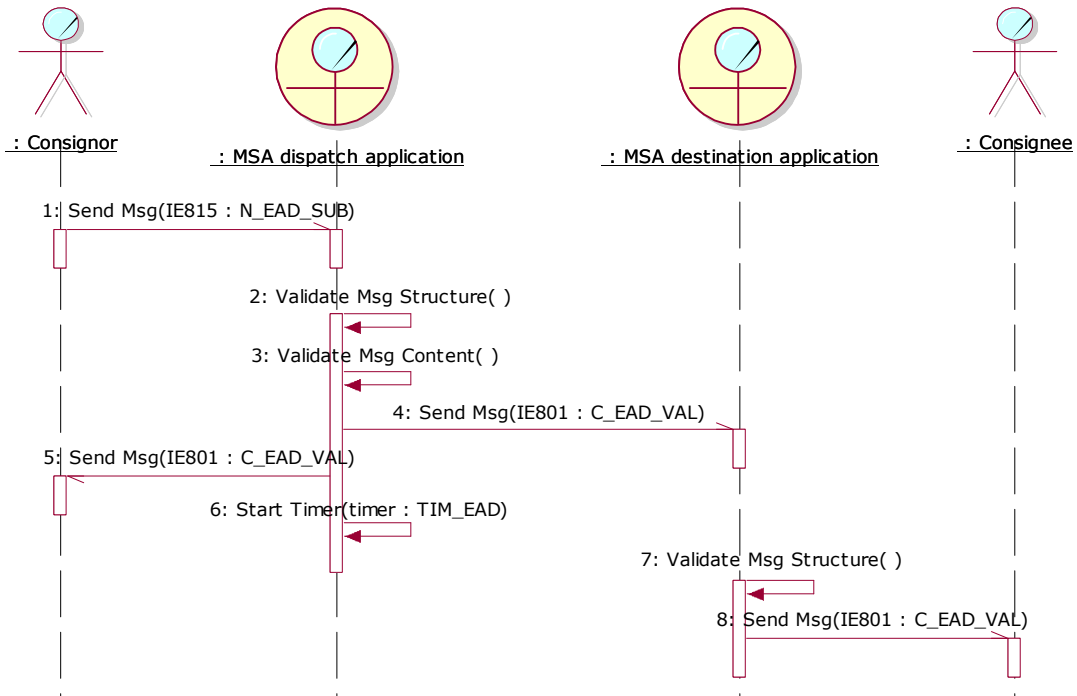


Figure 4: TSD - Origin is a tax warehouse & destination is known

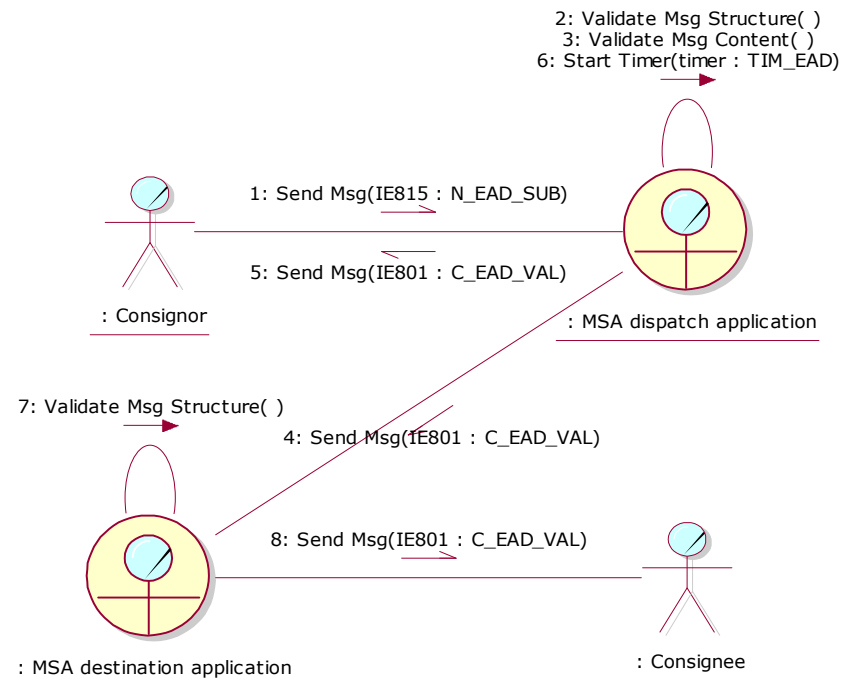


Figure 5: CLD - Origin is a tax warehouse & destination is known

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.1.1.2 Origin is tax warehouse and the destination is unknown

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The origin of the movement is a tax warehouse and the destination of the movement is unknown in accordance with Article 22 of Directive 2008/118/EC.

The MSA dispatch application upon receiving the draft e-AD performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application assigns an ARC to the e-AD (the structure of ARC and the Check Digit algorithm are defined in FESS Appendix B [A1]) and creates a validated e-AD (IE801: C_EAD_VAL) with sequence number "1" that disseminates to the Consignor. The validated e-AD (IE801: C_EAD_VAL) is stored by the MSA dispatch application in the "Accepted" state at the MSA of Dispatch. Finally, the timer TIM_EAD is initiated to expire at the expected end of movement (i.e. date of dispatch plus journey time) and the timer TIM_FDF is initiated to expire at the limit date for filling the destination fields.

The scenario of the submission and registration of an e-AD having a tax warehouse as origin and an unknown movement destination is depicted in Figure 6 and Figure 7:

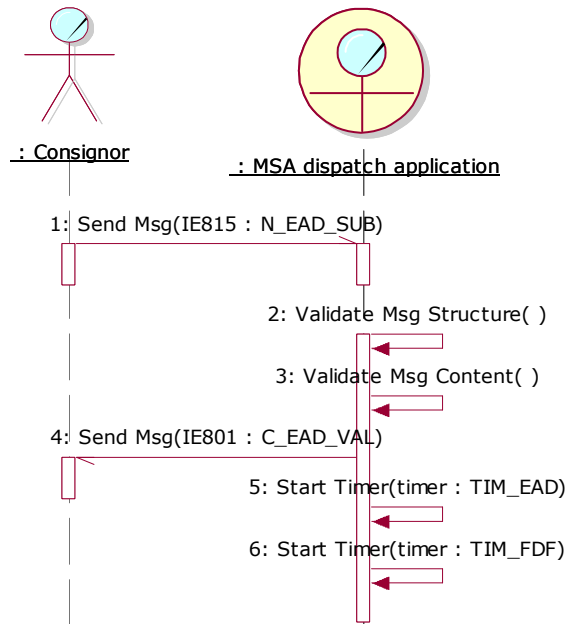


Figure 6: TSD - Origin is a tax warehouse & destination is unknown

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

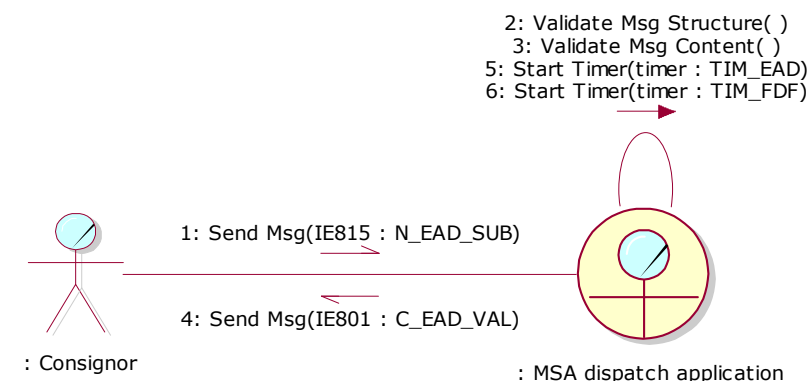


Figure 7: CLD - Origin is a tax warehouse & destination is unknown

III.I.1.2 Origin is import

This section describes message exchange protocol for the submission and registration of an e-AD having a place of importation as origin, having a registered consignor as the consignor and having the Customs office where the import procedure has been completed as the Place of Dispatch.

The scenarios of this section are optional since the interconnection between the EMCS and the Customs procedures depend on national provisions. The “Customs” business worker should be considered as either the Customs office of import or the Customs application of import. Each Member State may optionally implement support for automatic message exchanges between the MSA dispatch application and the Customs application of import.

The exceptional cases where the draft e-AD (IE815: N_EAD_SUB) submitted by the consignor is found to be invalid are described in Sub-Section III.VI.

III.I.1.2.1 Origin is import and the e-AD is inconsistent with the import data

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The origin of the movement is import.

The MSA dispatch application upon receiving the draft e-AD performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application forwards the draft e-AD (IE815: N_EAD_SUB) to Customs office and/or application to confirm consistency with the import data. The Customs office and/or application has found inconsistencies and notifies the MSA dispatch application with a rejection message (IE839: C_CUS_REJ) and the MSA dispatch application forwards the message to the Consignor.

The scenario of the submission and registration of an e-AD having a place of import as origin and being inconsistent with the import data is depicted in Figure 8 and Figure 9:

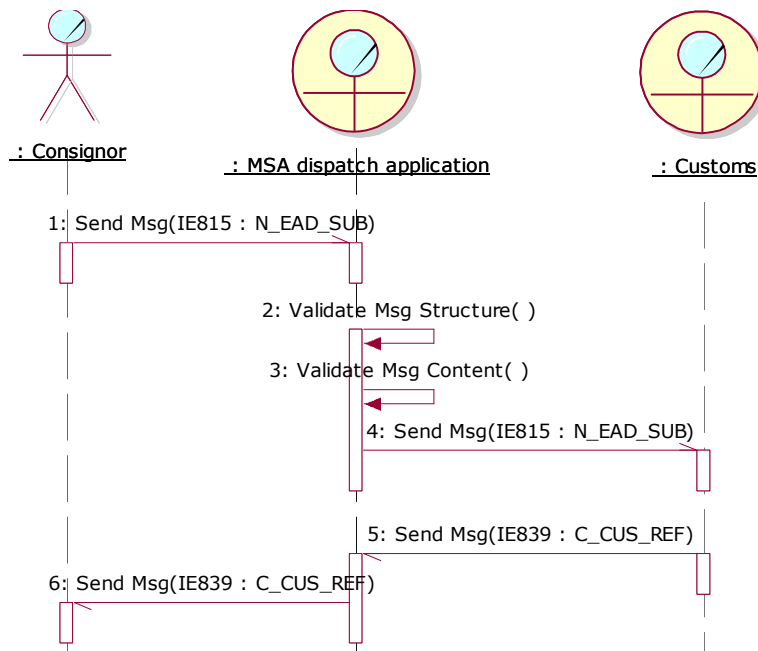


Figure 8: TSD - Origin is import and the e-AD is inconsistent with the import data

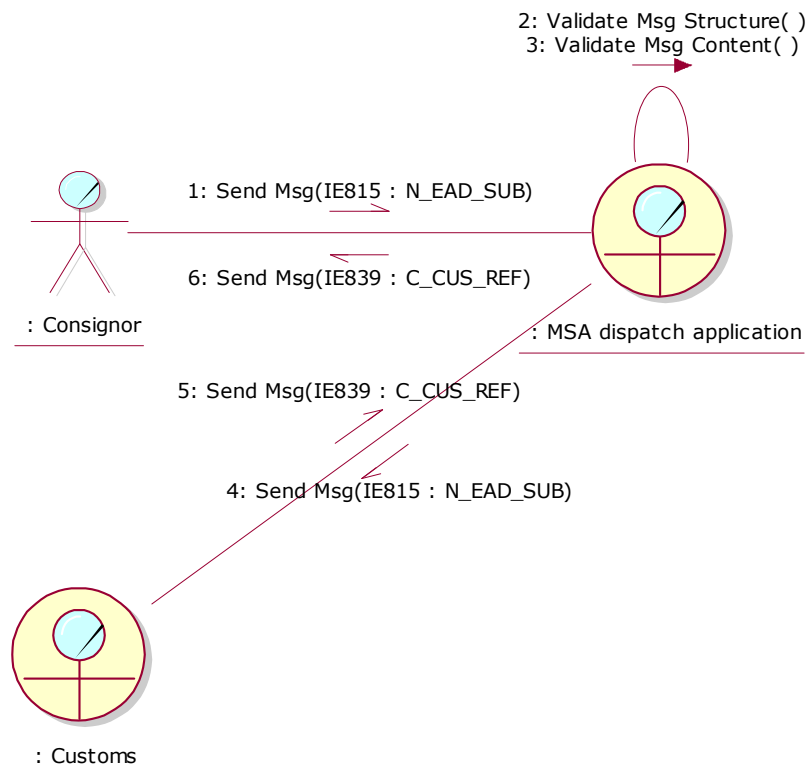


Figure 9: CLD -Origin is import and the e-AD is inconsistent with the import data

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.1.2.2 Origin is import and the destination is known

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The origin of the movement is import and the destination of the movement is known (i.e. the destination is not unknown nor it is for export). This scenario begins with the same sequence as the above scenario (III.I.1.2.1), only in this case the consistency with the import data is confirmed thus the MSA dispatch application and the validated e-AD is registered.

In particular, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The MSA dispatch application upon receiving the draft e-AD performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application forwards the draft e-AD to Customs office and/or application to confirm consistency with the import data. The Customs office and/or application has not found any inconsistencies and notifies the MSA dispatch application with an acceptance message (IE815: N_EAD_SUB).

The MSA dispatch application assigns an ARC to the e-AD (the structure of ARC and the Check Digit algorithm are defined in FESS Appendix B [A1]) and creates a validated e-AD (IE801: C_EAD_VAL) with sequence number "1" that disseminates to the MSA destination application and to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to "Accepted" and the timer TIM_EAD is initiated to expire at the expected end of movement (i.e. date of dispatch plus journey time).

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the MSA destination application stores the e-AD and sets the state of the e-AD at MSA of Destination to "Accepted". Finally, the MSA destination application forwards the e-AD (IE801: C_EAD_VAL) to the Consignee.

The scenario of the submission and registration of an e-AD having a place of import as origin and a known movement destination is depicted in Figure 10 and Figure 11:

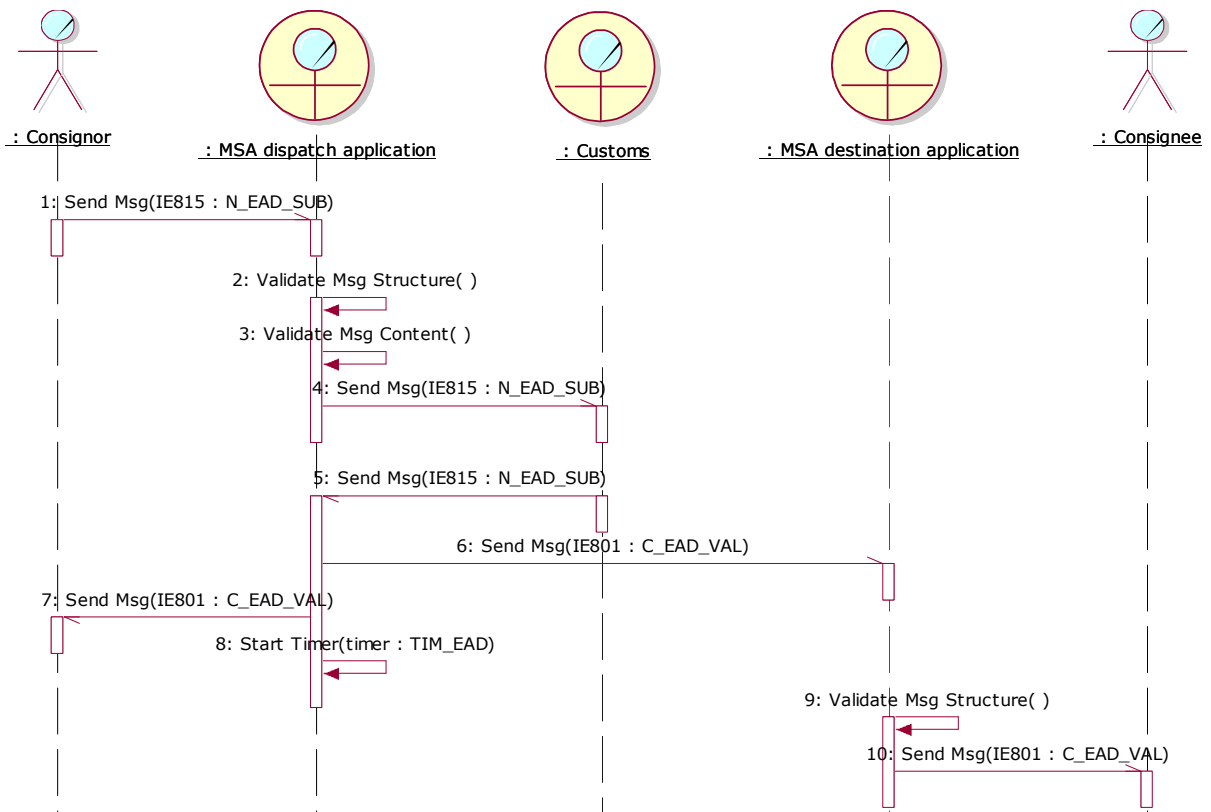


Figure 10: TSD - Origin is import and the destination is known

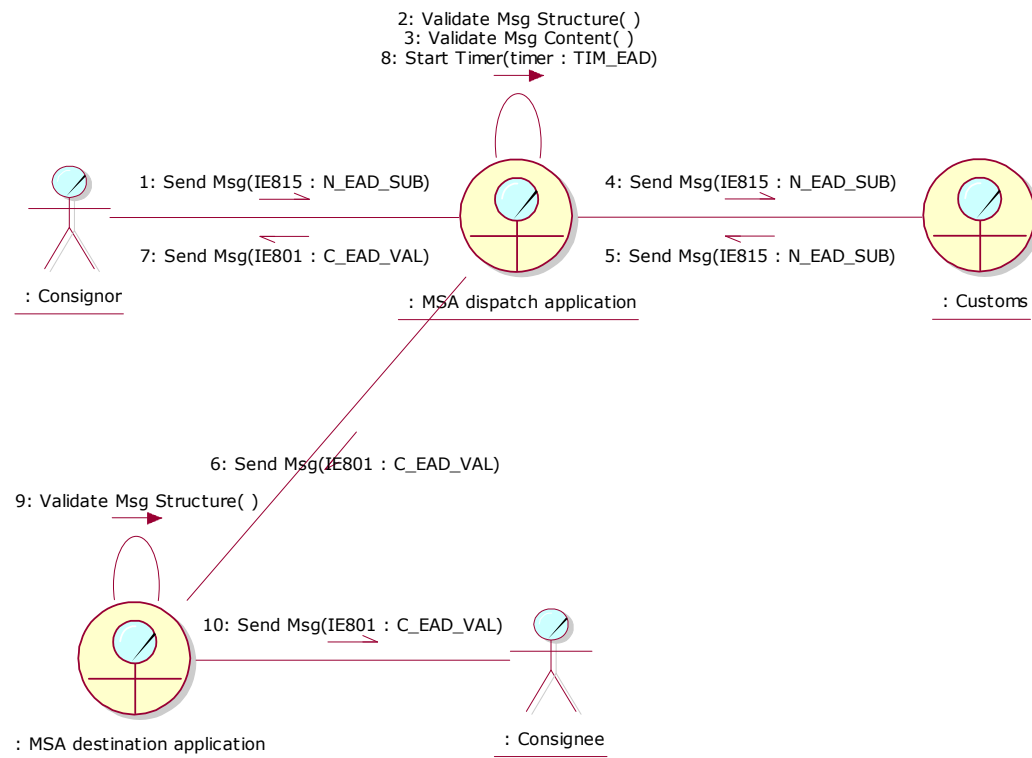


Figure 11: CLD - Origin is import and the destination is known

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.1.2.3 Origin is import and the destination is unknown

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The origin of the movement is import and the destination of the movement is unknown in accordance with Article 22 of Directive 2008/118/EC. This scenario begins with the same sequence as the above scenario (III.I.1.2.1), only in this case the consistency with the import data is confirmed thus the MSA dispatch application and the validated e-AD is registered.

In particular, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application. The MSA dispatch application upon receiving the draft e-AD performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application forwards the draft e-AD to Customs office and/or application to confirm consistency with the import data. The Customs office and/or application has found the draft e-AD (IE815: N_EAD_SUB) to be consistent with the import data and notifies the MSA dispatch application with an acceptance message (IE815: N_EAD_SUB).

The MSA dispatch application assigns an ARC to the e-AD (the structure of ARC and the Check Digit algorithm are defined in FESS Appendix B [A1]) and creates a validated e-AD (IE801: C_EAD_VAL) with sequence number “1” that disseminates to the Consignor. The validated e-AD (IE801: C_EAD_VAL) is stored by the MSA dispatch application in the “Accepted” state at the MSA of Dispatch. Finally, the timer TIM_EAD is initiated to expire at the expected end of movement (i.e. date of dispatch plus journey time) and the timer TIM_FDF is initiated to expire at the limit date for filling the destination fields.

The scenario of the submission and registration of an e-AD having a place of import as origin and an unknown movement destination is depicted in Figure 12 and Figure 13:

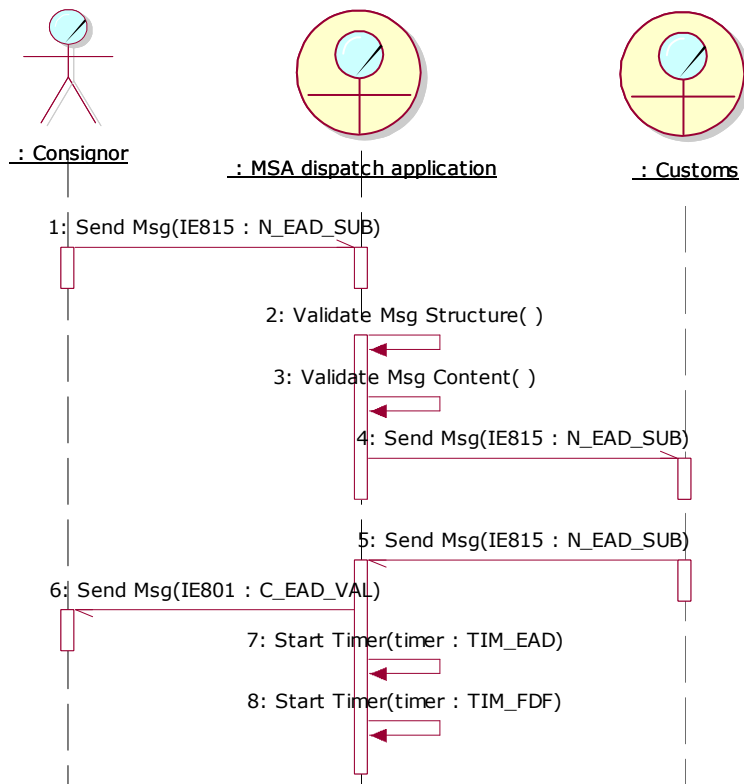


Figure 12: TSD - Origin is import and the destination is unknown

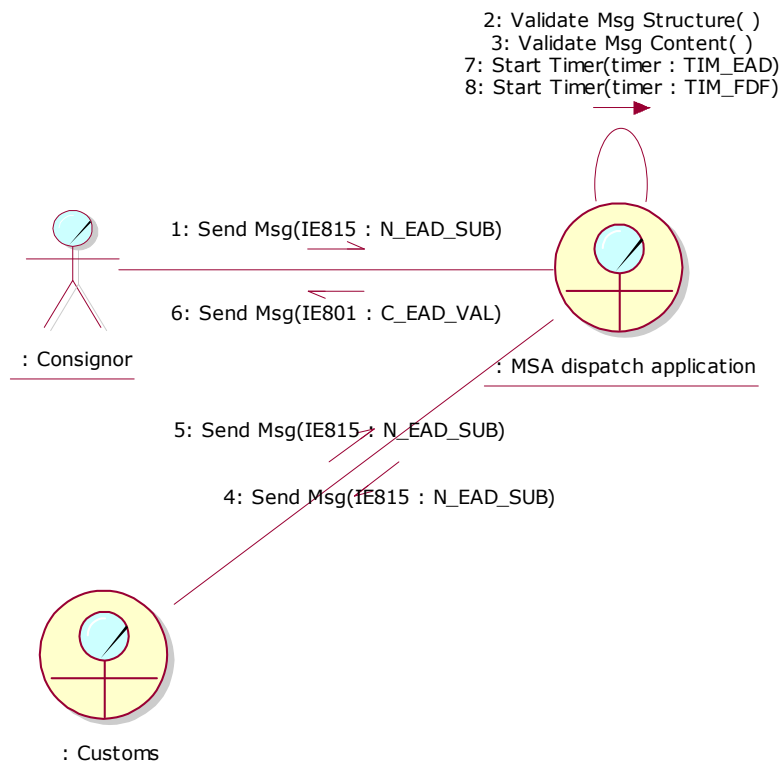


Figure 13: CLD - Origin is import and the destination is unknown

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.2 Alert or rejection of an e-AD (UC2.07)

The Consignee may submit an alert or rejection (IE819: C_REJ_DAT) before the goods arrive at the destination to notify the involved actors. The successful completion of the scenarios in this section cause state transition for the concerned e-AD to “*Accepted*” or “*Rejected*” when Consignee alerts or rejects the e-AD, respectively. The state transition pertains to both the MSA of Dispatch and the MSA of Destination.

The scenarios of this use case are initiated by the Consignee, hence the destination of the concerned e-AD cannot be unknown since in this case the movement has no Consignee. Moreover, the scenarios of this use case require the state of the concerned e-AD to be in the “*Accepted*” state.

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.II.1.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.6.2	Change of Destination (UC2.05)	Accepted
III.II.2.8	Export Operation at Office of Export and movement not released by Customs followed by change of destination	Accepted
III.II.3.3.2	Change of Destination (UC2.05)	Accepted
III.II.3.6.2	Change of Destination (UC2.05)	Accepted

Table 8: Scenarios that must have been previously completed in order to submit an alert or rejection

The different scenarios of this use case are determined by the indicator of the alert or rejection, which can either signify an alert or rejection:

- e-AD alerted (see III.I.4.1); and
- e-AD rejected (see III.I.4.2).

Both scenarios begin essentially with the Consignee submitting an alert or rejection (IE819: C_REJ_DAT) to the MSA destination application that subsequently validates the structure and the message content. The exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

III.I.2.1 e-AD alerted

According to this scenario, the Consignee submits an alert (IE819: C_REJ_DAT) to the MSA destination application. The Consignee has provided the ARC and the last sequence number of the movement that he/she alerts, which is in the “*Accepted*” for both the MSA of Dispatch and the MSA of Destination.

The MSA destination application upon receiving the alert (IE819: C_REJ_DAT) performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA destination does not alter the state of the concerned e-AD and forwards the alert (IE819: C_REJ_DAT) to the MSA dispatch application. Finally, the MSA destination application sends back the validated alert to the Consignee as a confirmation.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon the reception of the alert (IE819: C_REJ_DAT) the MSA dispatch application does not alter the state of the concerned e-AD and forwards the alert (IE819: C_REJ_DAT) to the Consignor.

The scenario of the submission and registration of an alert is depicted in Figure 14 and Figure 15:

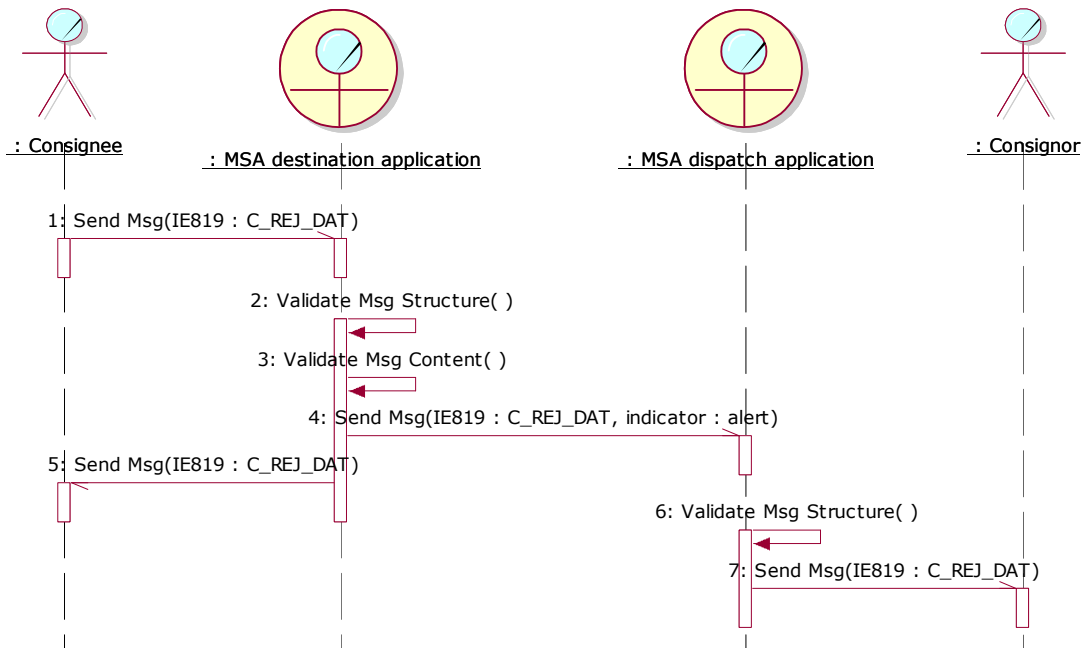


Figure 14: TSD - e-AD alerted

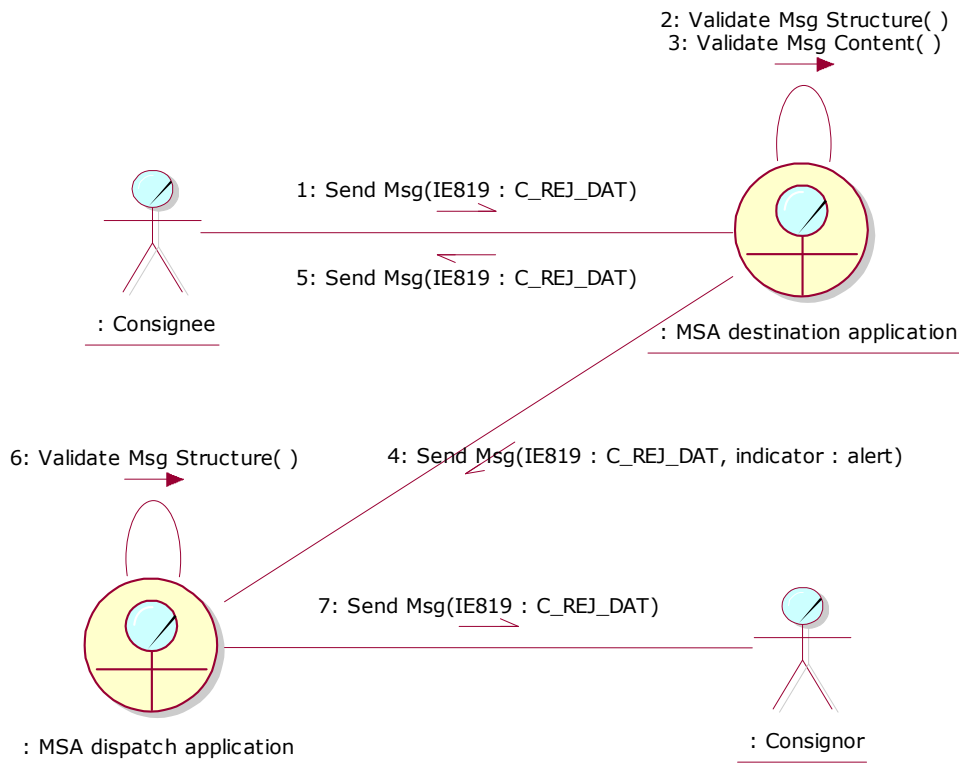


Figure 15: CLD - e-AD alerted

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.2.2 e-AD rejected

According to this scenario, the Consignee submits a rejection (IE819: C_REJ_DAT) to the MSA destination application. The Consignee has provided the ARC of the movement and the last sequence number that he/she rejects, which is in the “Accepted” for both the MSA of Dispatch and the MSA of Destination.

The MSA destination application upon receiving the rejection (IE819: C_REJ_DAT) performs the relevant validations, which pass successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA destination application sets the state of the movement at the MSA of Destination to “Rejected” and forwards the rejection (IE819: C_REJ_DAT) to the MSA dispatch application. Finally, the MSA destination application sends back the validated rejection to the Consignee as a confirmation.

Upon the reception of the rejection (IE819: C_REJ_DAT) the MSA dispatch application sets the state of the movement at the MSA of Dispatch to “Rejected” and forwards the rejection to the Consignor. The Consignor is expected to change the e-AD destination, split the e-AD, or cancel the e-AD as a response to the rejection and the MSA dispatch application initiates the TIM_CHS timer, which once elapsed, will send a reminder to the Consignor.

Please note that the “Rejected” state is not a final state and the Consignor is expected to take actions such that the e-AD will finally reach a final state. The expected actions from the Consignor and the timer TIM_CHS expiration are described in chapter III.I.8.

The scenario of the submission and registration of a rejection is depicted in Figure 16 and Figure 17:

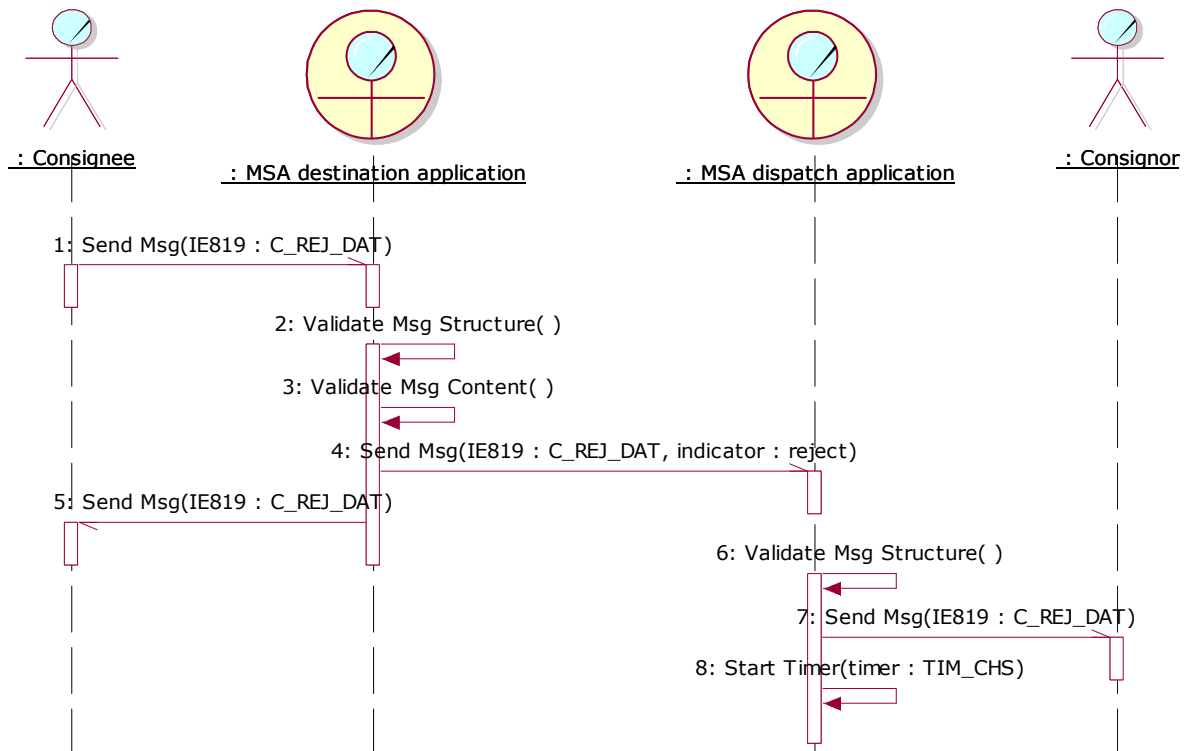


Figure 16: TSD - e-AD rejected

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

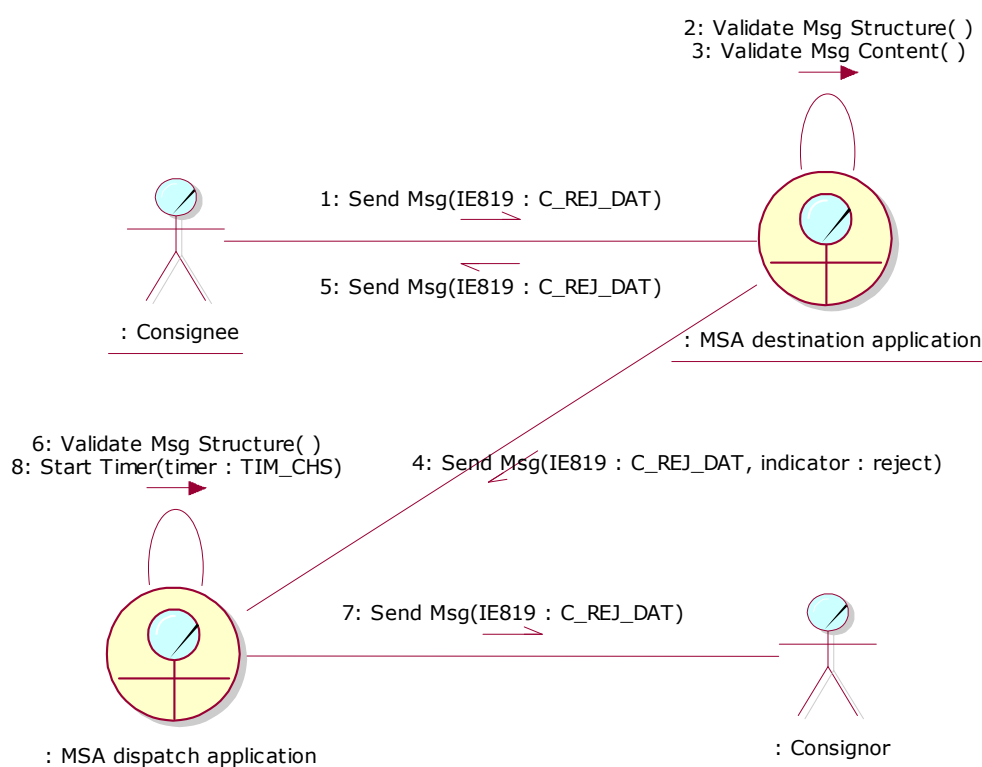


Figure 17: CLD - e-AD rejected

III.I.3 Cancellation of an e-AD by the Consignor (UC2.10)

If a *consignment* has *NOT* left the *Place of Dispatch*, then the Consignor may submit an e-AD cancellation. A consignment is considered to have left the place of Dispatch in the following cases:

- The Consignor has issued at least one change of destination (IE813) for the e-AD;
- The e-AD has resulted from a splitting operation (IE825) that is the e-AD is a downstream e-AD;
- The e-AD has been submitted in deferred mode; electronic recovery of fallback AD may only occur after the physical dispatch of the goods (for electronic recovery of fallback AD forms see FRS [A12]);
- The e-AD is in the exporting state and the export operations take place at the Office of Export (Submission Message Type of the draft e-AD is "Standard submission").

The successful completion of the scenario in this section causes state transition for the concerned e-AD to “*Cancelled*” pertaining to both the MSA of Dispatch and the MSA of Destination.

The scenario of this use case requires the concerned e-AD to be in “*Accepted*” or “*Rejected*” state.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.I.2.2	e-AD rejected	Rejected

Table 9: Scenarios that must have been previously completed in order to submit an e-AD cancellation

This scenario assumes that all validations of the incoming messages pass successfully; the exceptional cases are described in Sub-Section III.VI.

Anytime before the actual dispatch of goods, the Consignor may send a cancellation message (IE810: C_CAN_DAT) to the MSA dispatch application concerning an e-AD in the “Accepted” or “Rejected” or “Exporting” (only for the case that the draft e-AD has been submitted for “Local clearance at export”) or “Accepted for Export” state.

The MSA dispatch application receives the draft e-AD cancellation for validation (IE810: C_CAN_DAT). Upon successful validation of the incoming message, the MSA dispatch application updates the state of the e-AD to “Cancelled” and forwards the cancellation notification (IE810: C_CAN_DAT) to the MSA destination application and back to the Consignor.

If the timer associated with the cancelled e-AD (TIM_EAD) has already expired at the limit date, the MSA dispatch application resets the flag that has been raised locally at expiration time. In the opposite case, if the timer (TIM_EAD) associated with the cancelled e-AD is still running, the MSA dispatch application stops it.

The MSA destination application receives the cancellation notification message for validation (IE810: C_CAN_DAT). Upon successful validation of the incoming message, the MSA destination application changes the state of the e-AD to “Cancelled” and forwards the cancellation notification (IE810: C_CAN_DAT) to the Consignee.

The cancellation of an e-AD is always a final operation and the movement state at the MSA dispatch and destination applications is updated to “Cancelled”, which is a final state.

The scenario to cancel an e-AD is depicted in Figure 18 and Figure 19:

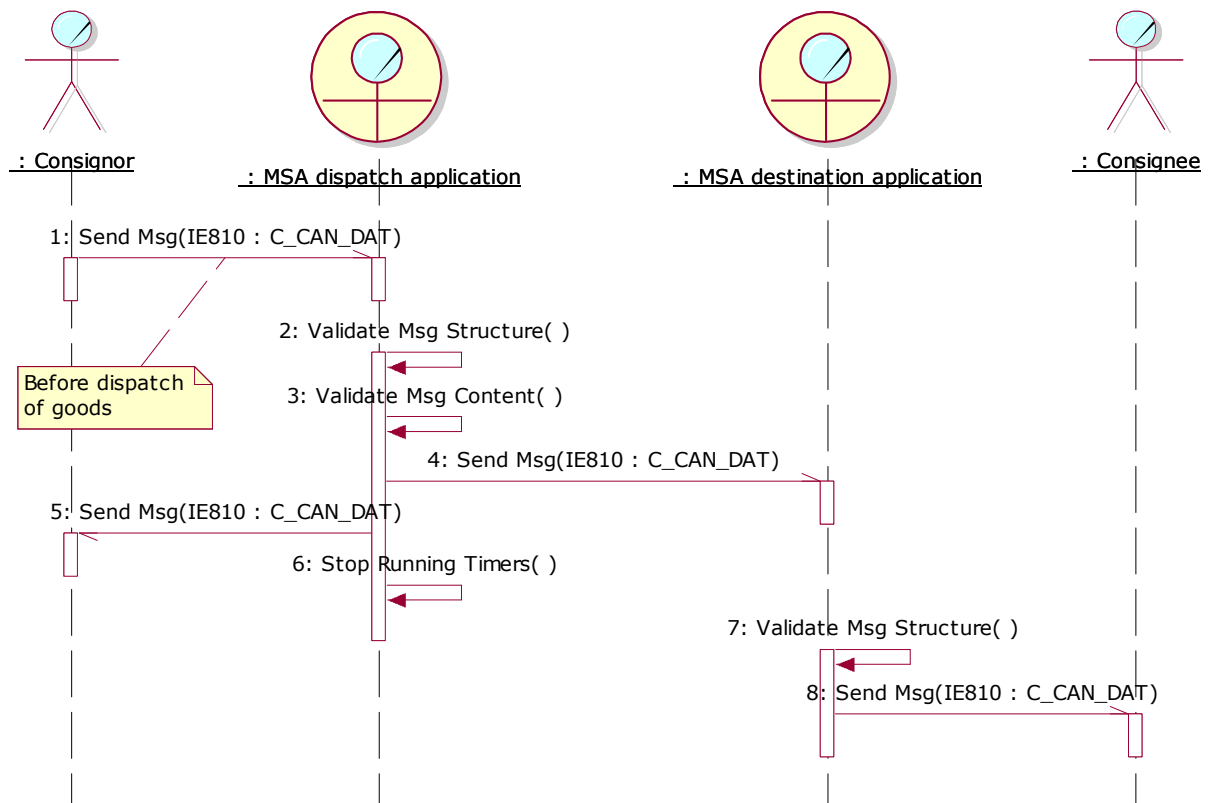


Figure 18: TSD - Cancellation of an e-AD by the Consignor (UC2.10)

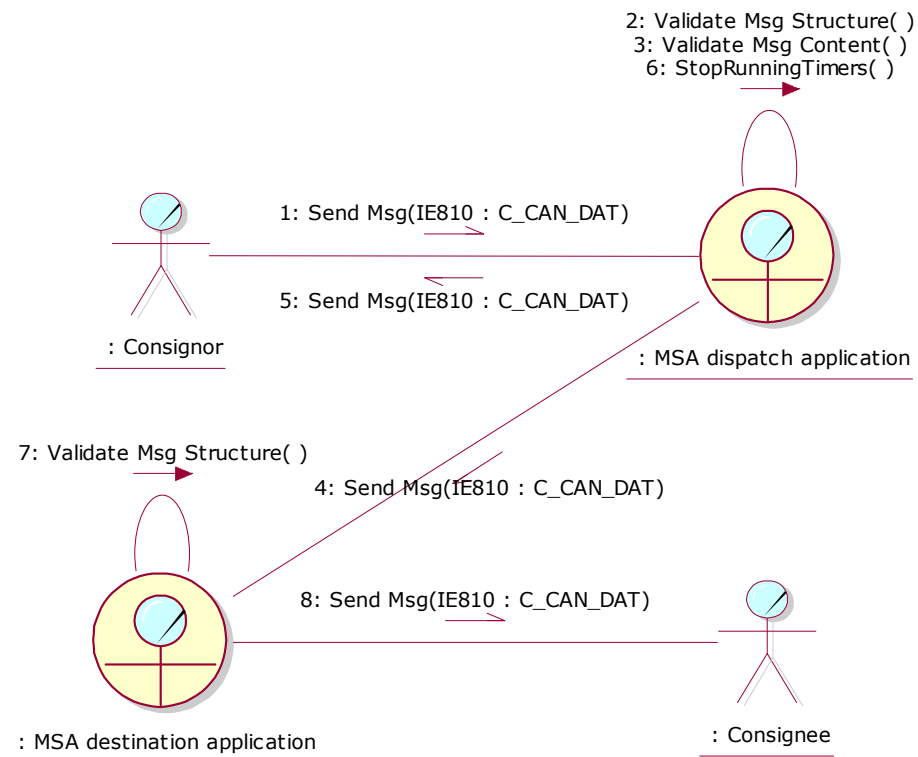


Figure 19: CLD - Cancellation of an e-AD by the Consignor (UC2.10)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.4 Submission of Report of Receipt (UC2.06)

When the goods arrive at their destination, the Consignee acknowledges the receipt of goods by submitting a Report of Receipt (RoR) (IE818: C_DEL_DAT) to notify the involved actors. The Consignee includes in the draft Report of Receipt (IE818: C_DEL_DAT) the ARC and the last sequence number of the ARC. The successful completion of the scenarios in this section cause state transition for the concerned e-AD to “*Delivered*”, “*Refused*”, or “*Partially refused*”.

The scenarios of this use case require the state of the concerned e-AD to be in the “*Accepted*” state. In addition, the receipt of goods may occur when the e-AD is in the “*Exporting*” state in which case the scenarios for the exportation of goods apply described in Sub-Section III.II Export Scenarios.

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.II.1.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.6.2	Change of Destination (UC2.05)	Accepted
III.II.2.8	Export Operation at Office of Export and movement not released by Customs followed by change of destination	Accepted
III.II.3.3.2	Change of Destination (UC2.05)	Accepted
III.II.3.6.2	Change of Destination (UC2.05)	Accepted
III.I.2.1	e-AD alerted	Accepted

Table 10: Scenarios that must have been previously completed in order to submit a Report of Receipt

The different scenarios of this use case are determined by the following value categories of the RoR conclusion:

- delivery accepted (see III.I.4.1);
- delivery refused (see III.I.4.2); and
- delivery partially refused (see III.I.4.3).

Two additional value categories of the RoR conclusion are possible pertaining to the scenarios for the exportation of goods. These scenarios are not included in this section and are described in Sub-Section III.II Export Scenarios.

All scenarios begin essentially with the Consignee submitting a Report of Receipt (IE818: C_DEL_DAT) to the MSA destination application that subsequently validates the structure and the message content. The exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.4.1 Delivery Accepted

According to this scenario, the Consignee submits a draft Report of Receipt (IE818: C_DEL_DAT) indicating delivery acceptance to the MSA destination application that subsequently validates successfully the structure and the message content (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). Then the MSA destination application changes the state of the e-AD at MSA of Destination to “*Delivered*”, which is a final state, and forwards the validated Report of Receipt (IE818: C_DEL_DAT) to the MSA dispatch application. Finally, the MSA destination application sends back the validated RoR to the Consignee as a confirmation.

Upon the reception of delivery notification message (IE818: C_DEL_DAT), the MSA dispatch application validates successfully the received message and changes the state of the e-AD to “*Delivered*”, which is a final state. In addition, the MSA dispatch application forwards the delivery notification (IE818: C_DEL_DAT) to the Consignor to inform him/her for the acceptance of delivery and in series the discharge of the movement. Finally, if the TIM_EAD timer has not expired, the MSA dispatch application stops it otherwise it resets the flag raised locally at its expiration.

The scenario with the acceptance of delivery is depicted in Figure 20 and Figure 21:

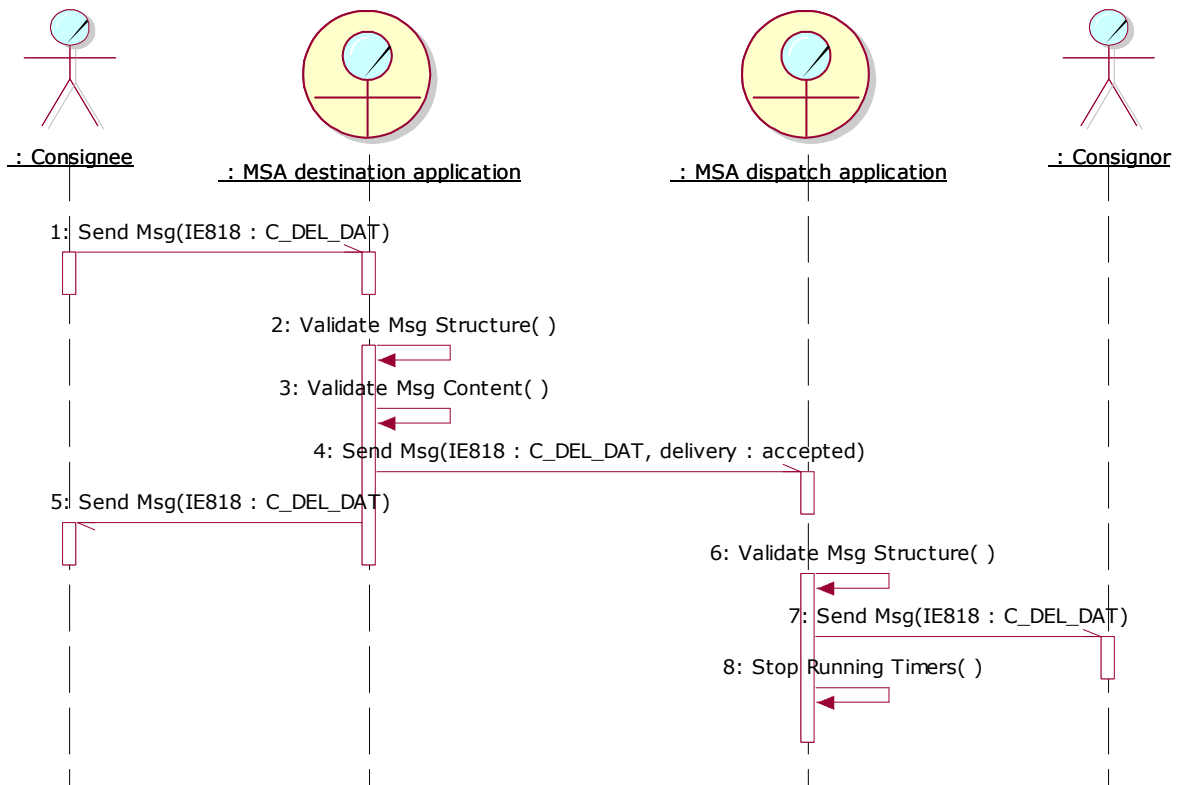


Figure 20: TSD - Submission of e-AD of which delivery is “Accepted” (with or without shortages)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

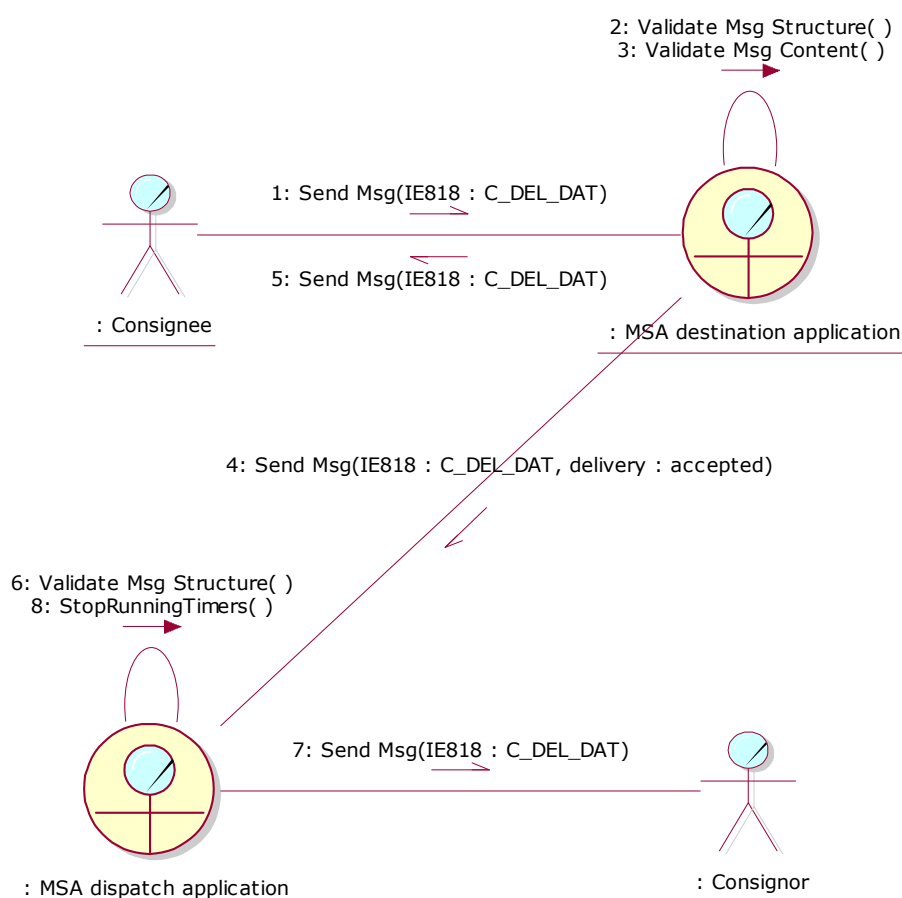


Figure 21: CLD - Submission of e-AD of which delivery is “Accepted” (with or without shortages)

III.I.4.2 Delivery Refused

If the Consignee submits a draft Report of Receipt (IE818: C_DEL_DAT) indicating refusal of delivery to the MSA destination application, the latter validates successfully the structure and the message content (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). Then the MSA destination application changes the state of the e-AD at MSA of Destination to “*Refused*” and forwards the validated Report of Receipt (IE818: C_DEL_DAT) to the MSA dispatch application. Finally, the MSA destination application sends back the validated RoR to the Consignee as a confirmation.

Upon the reception of the refusal of delivery notification message (IE818: C_DEL_DAT), the MSA dispatch application validates successfully the received message and changes the state of the e-AD to “*Refused*”. In addition, the MSA dispatch application forwards the delivery notification to the Consignor. Finally, the MSA dispatch application starts the timer TIM_CHS, which once elapsed, will send a reminder to the Consignor.

Please note that the “*Refused*” state is not a final state and the Consignor is expected to take actions such that the e-AD will finally reach a final state. The expected actions from the Consignor and the timer TIM_CHS expiration are described in chapter III.I.8.

The scenario with the refusal of delivery is depicted in Figure 22 and Figure 23:

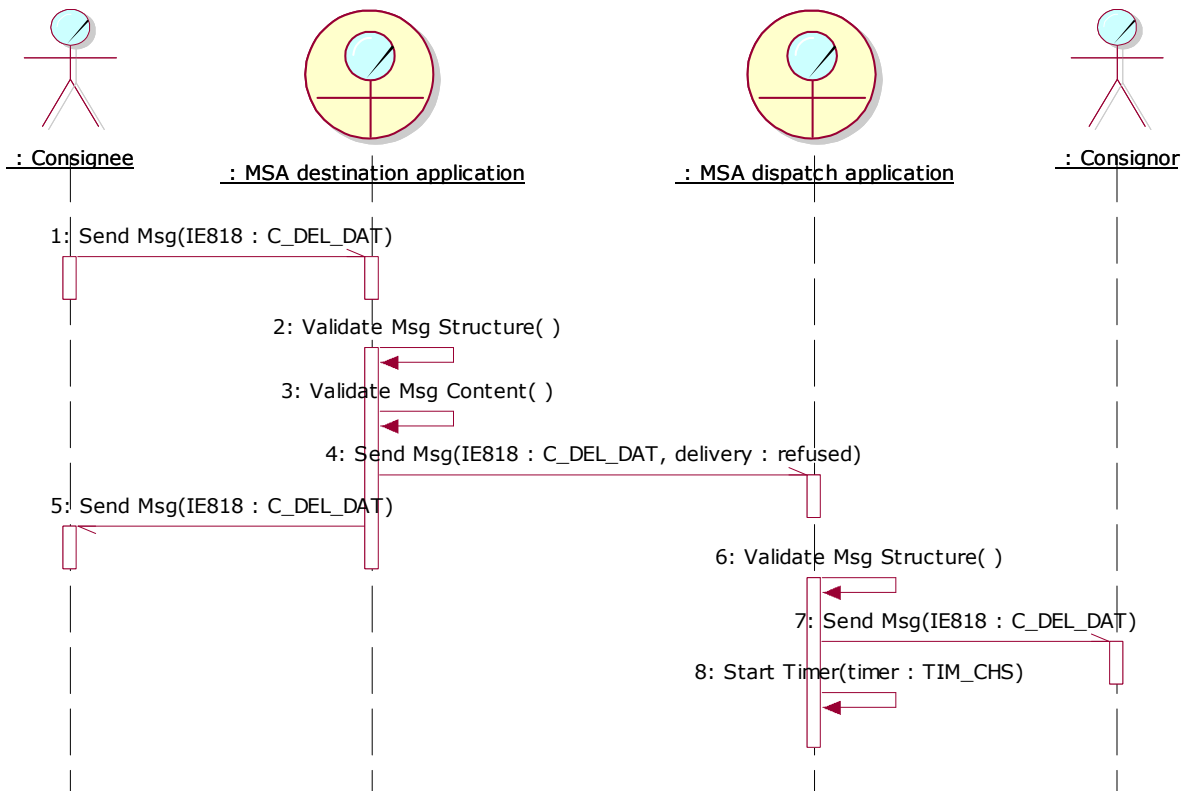


Figure 22: TSD - Submission of e-AD of which delivery is “Refused”

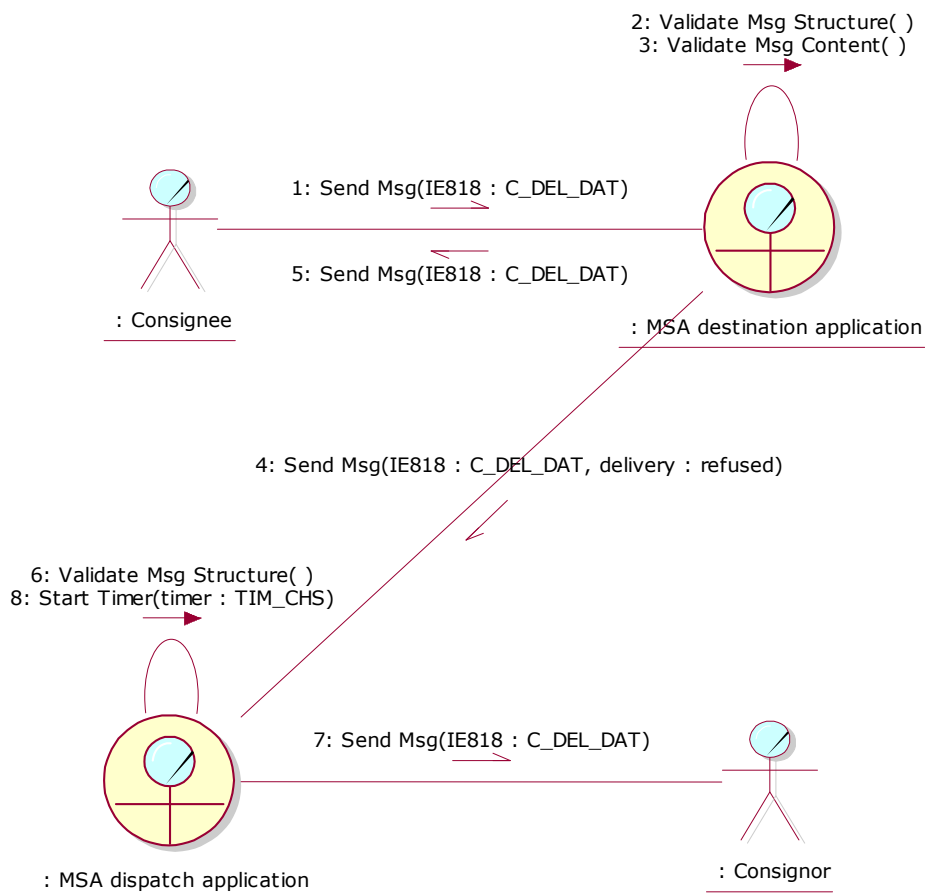


Figure 23: CLD - Submission of e-AD of which delivery is “Refused”

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.4.3 Delivery Partially Refused

If the Consignee submits a draft Report of Receipt (IE818: C_DEL_DAT) indicating partial refusal of delivery to the MSA destination application, the latter validates successfully the structure and the message content (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). Then the MSA destination application changes the state of the e-AD at MSA of Destination to “*Partially Refused*” and forwards the validated Report of Receipt (IE818: C_DEL_DAT) to the MSA dispatch application. Finally, the MSA destination application sends back the validated RoR to the Consignee as a confirmation.

Upon the reception of the partial refusal of delivery notification message (IE818: C_DEL_DAT), the MSA dispatch application validates successfully the received message and changes the state of the e-AD to “*Partially Refused*”. In addition, the MSA dispatch application forwards the delivery notification to the Consignor. Finally, the MSA dispatch application starts the timer TIM_CHS, which once elapsed, will send a reminder to the Consignor.

Please note that the “*Partially Refused*” state is not a final state and the Consignor is expected to take actions such that the e-AD will finally reach a final state. The expected actions from the Consignor and the timer TIM_CHS expiration are described in chapter III.I.8.

The scenario with the partial refusal of delivery is depicted in Figure 24 and Figure 25:

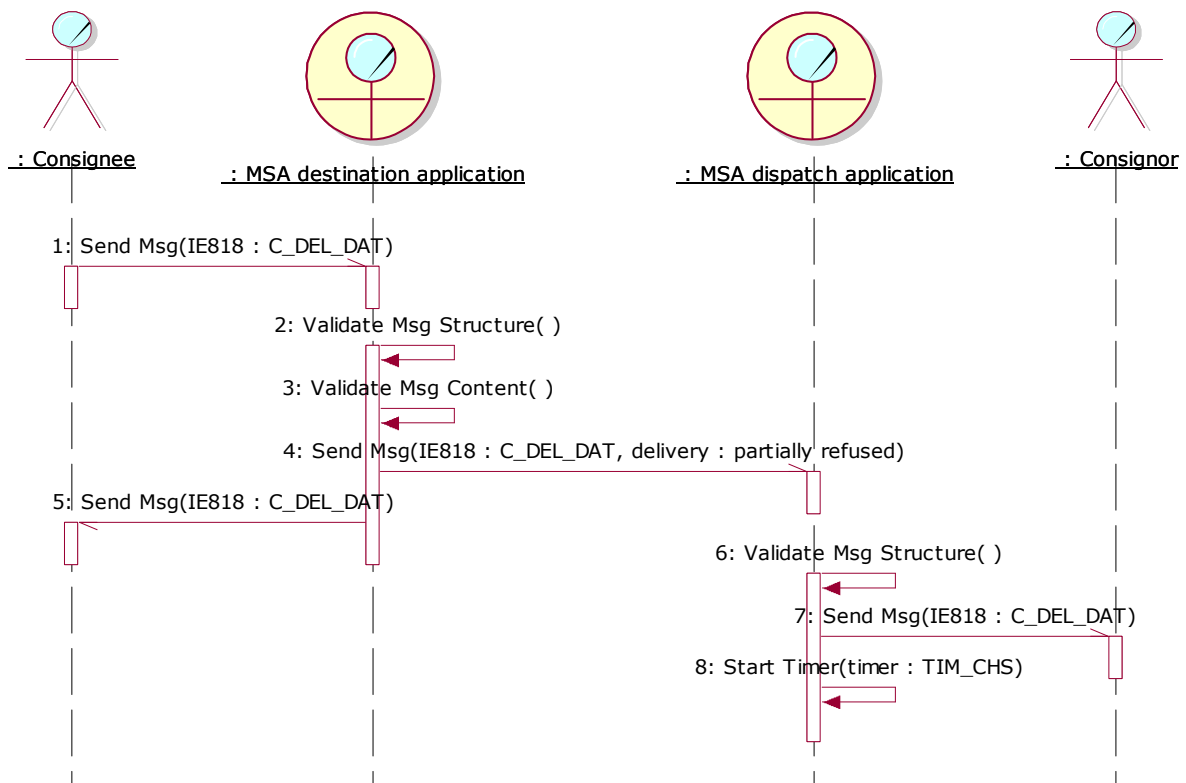


Figure 24: TSD - Submission of e-AD of which delivery is “Partially Refused”

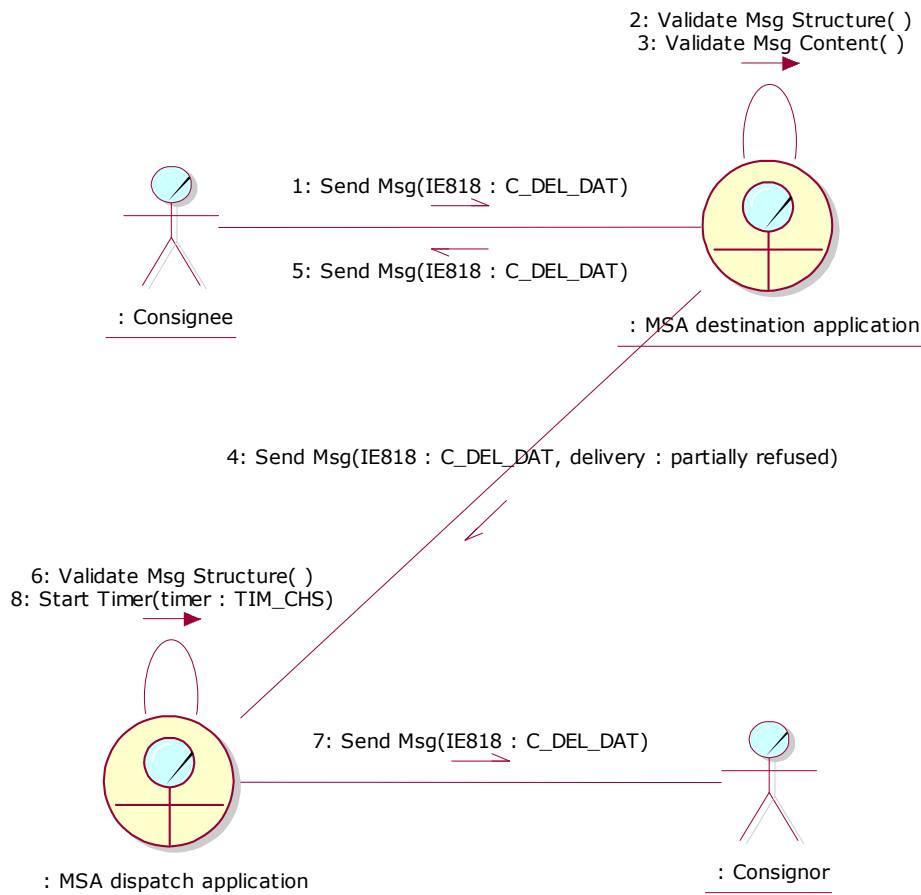


Figure 25: CLD - Submission of e-AD of which delivery is “Partially Refused”

III.I.5 Change of Destination (UC2.05)

The Consignor may change the destination of an e-AD. The successful completion of the scenario in this section causes state transition for the concerned e-AD to “Accepted” pertaining to both the MSA of Dispatch and the MSA of Destination. The new destination can be a tax warehouse or the premises of a registered Consignee or of a temporary registered Consignee or a place of direct delivery or export.

The scenarios of this use case require the state of the concerned e-AD to be in the “Accepted”, “Refused”, “Partially refused”, or “Rejected” state. Moreover, the scenarios of this use case require the state of the concerned e-AD to be in the “Exporting” state in which case the scenarios for the exportation of goods apply described in Sub-Section III.II Export Scenarios.

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.II.1.2.3	Change of Destination (UC2.05)	Accepted
III.II.1.3.2	Change of Destination (UC2.05)	Accepted
III.II.2.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.3.2	Change of Destination (UC2.05)	Accepted
III.II.2.6.2	Change of Destination (UC2.05)	Accepted

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2.8	Export Operation at Office of Export and movement not released by Customs followed by change of destination	Accepted
III.II.3.3.2	Change of Destination (UC2.05)	Accepted
III.II.3.4.2	Change of Destination (UC2.05)	Accepted
III.II.3.6.2	Change of Destination (UC2.05)	Accepted
III.I.2.2	e-AD rejected	Rejected
III.I.4.2	Delivery Refused	Refused
III.I.4.3	Delivery Partially Refused	Partially refused

Table 11: Scenarios that must have been previously completed in order to submit a change of destination

The different scenarios of this use case are determined by the combinations of the former destination and the new destination. There are three possible change of destination scenarios, either:

- the MS of Destination changes: this implies that both Consignee and Place of Delivery change (see III.I.5.1); or
- the MS of Destination does not change but the Consignee (hence the Place of Delivery) changes (see III.I.5.4); or
- neither the MS of Destination nor the Consignee change, but only the Place of Delivery changes (see III.I.5.5).

A particular case of the first scenario (where MS of Destination changes) is when the Consignor changes the destination for return of goods. In this scenario the new destination is the Place of Dispatch, which implies that both Consignee and Place of Delivery change (see III.I.5.3).

Furthermore, the particular case of the first scenario where the Consignor diverts a movement back to a (former) MSA of Destination is also included as a change of both Consignee and Place of Delivery (see III.I.5.2).

All scenarios begin essentially with the consignor submitting an update message (IE813: C_UPD_DAT) to the MSA dispatch application that subsequently validates the structure and the message content. The exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

All the possible change of destination scenarios are described below.

III.I.5.1 Change MS of Destination

The Consignor initiates the change of destination process by submitting an update message (IE813: C_UPD_DAT) to change the MS of Destination. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The MSA dispatch application includes the last sequence number of the ARC incremented by 1 (PreviousSeqNo + 1) in the validated update message (IE813: C_UPD_DAT) and sends it to the former MSA destination application as well as to the Consignor as an acknowledgement.

In addition, the MSA dispatch application generates and sends to the new MSA of Destination an e-AD (IE801: C_EAD_VAL) that includes the following information:

- The ARC of the update message (IE813: C_UPD_DAT), which is the same as in the original e-AD;
- The sequence number of the ARC which is incremented by 1 (PreviousSeqNo + 1);
- The updated destination details (new Consignee and Place of Delivery), as declared in the update message (IE813: C_UPD_DAT).

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

Upon the reception of the updated e-AD (IE801: C_EAD_VAL), the new MSA destination application receives and validates the e-AD. Assuming that the validation passes successfully, the new MSA destination application sends the e-AD (IE801: C_EAD_VAL) to the new Consignee to inform him that he is the new Consignee of the movement. Moreover, the new MSA destination application sets the state of the movement to *“Accepted”*.

At the other side, the former MSA destination application validates the structure of the received update message (IE813: C_UPD_DAT). Assuming that the message structure validation passes successfully, the former MSA destination application sets the state of e-AD to *“Diverted”* and sends a notification message (IE803: C_EAD_NOT) with the same ARC and sequence number as in the update message (IE813: C_UPD_DAT) to the former Consignee to inform him that the movement has changed destination.

The scenario to change the MS of Destination is depicted in Figure 26 and Figure 27:

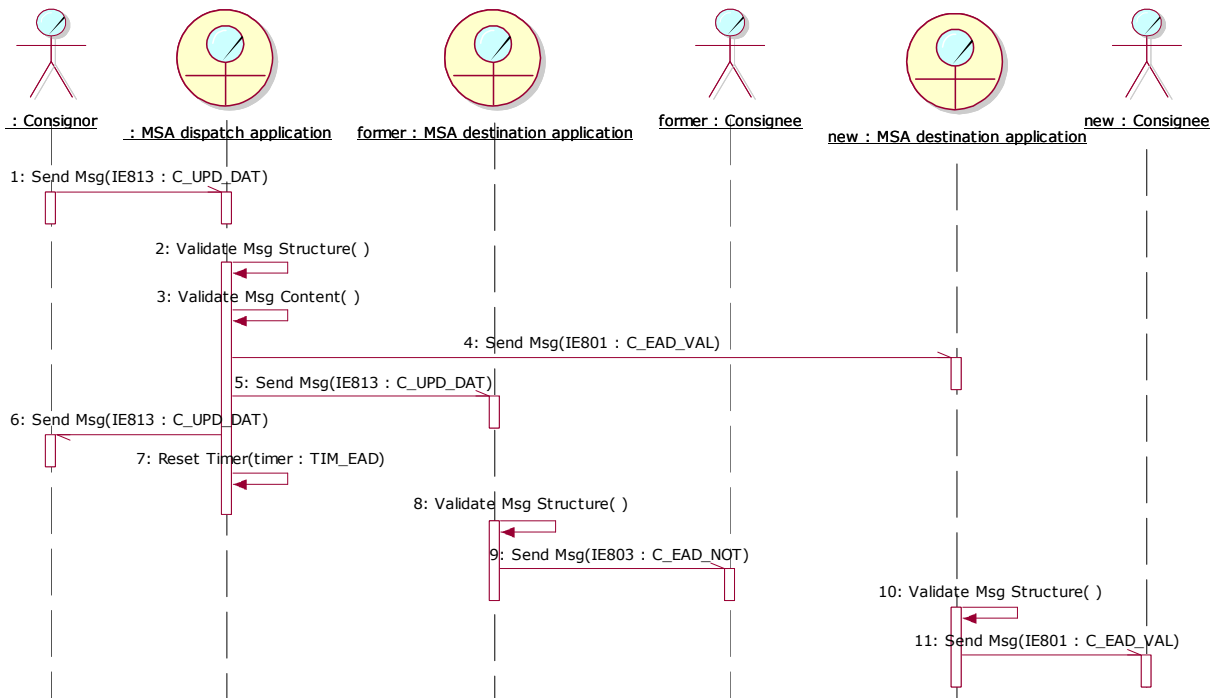


Figure 26: TSD - Change of MS of Destination

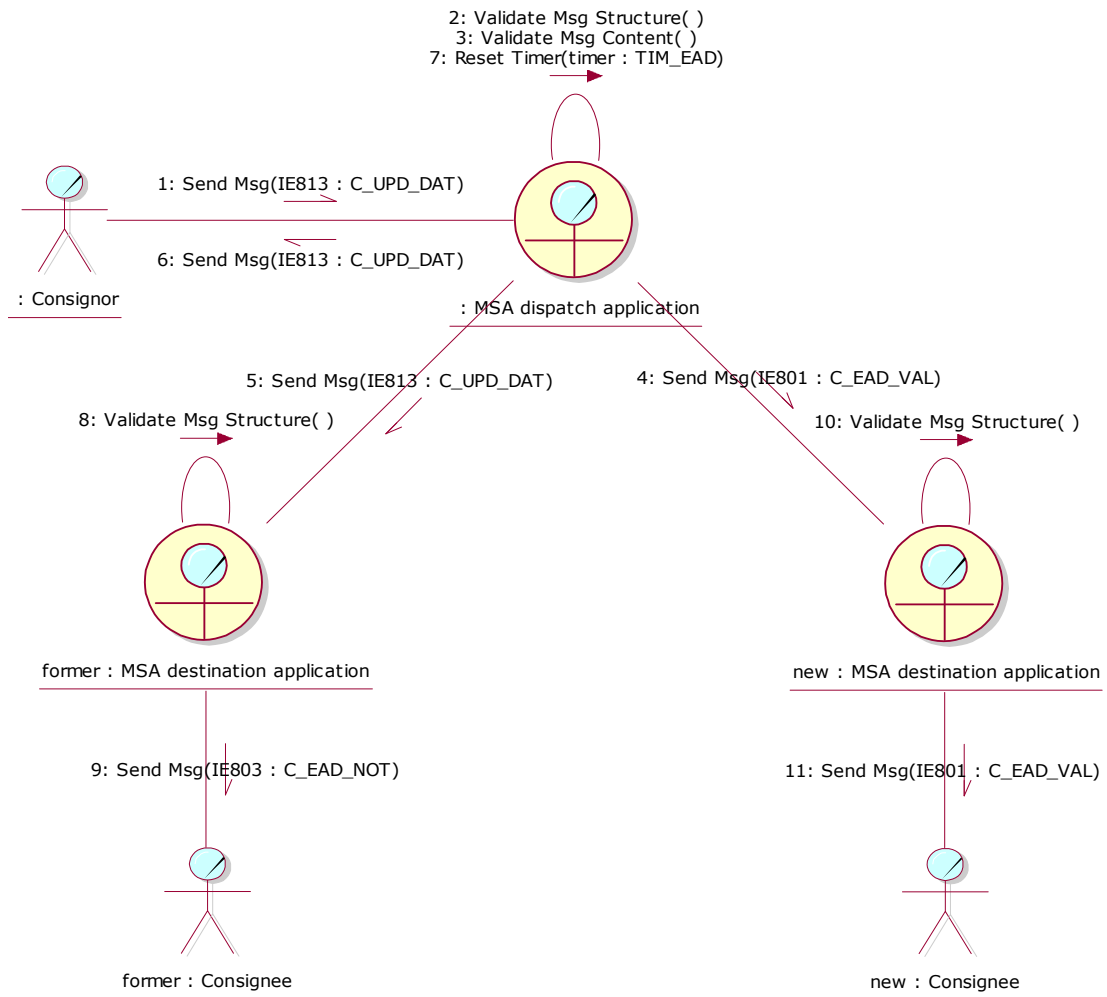


Figure 27: CLD - Change of MS of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.5.2 Change MS of Destination where the consignor diverts a movement back to a (former) MSA of Destination

The Consignor initiates the change of destination process by submitting an update message (IE813: C_UPD_DAT) to change the MS of Destination. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

The MSA dispatch application includes the last sequence number of the ARC incremented by 1 (PreviousSeqNo + 1) in the validated update message (IE813: C_UPD_DAT) and sends it to the former MSA destination application as well as to the Consignor as an acknowledgement.

In addition, the MSA dispatch application generates and sends to the new MSA of Destination an e-AD (IE801: C_EAD_VAL) that includes the following information:

- The ARC of the update message (IE813: C_UPD_DAT), which is the same as in the original e-AD;
- The sequence number of the ARC which is incremented by 1 (PreviousSeqNo + 1);
- The updated destination details (new Consignee and Place of Delivery), as declared in the update message (IE813: C_UPD_DAT).

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

Upon the reception of the updated e-AD (IE801: C_EAD_VAL), the new MSA destination application receives and validates the e-AD. Assuming that the validation passes successfully, the new MSA destination application sends the e-AD (IE801: C_EAD_VAL) to the new Consignee to inform him that he is the new Consignee of the movement. Moreover, the new MSA destination application sets the state of the movement to “Accepted”.

At the other side, the former MSA destination application validates the structure of the received update message (IE813: C_UPD_DAT). Assuming that the message structure validation passes successfully, the former MSA destination application sets the state of e-AD to “Diverted” and sends a notification message (IE803: C_EAD_NOT) with the same ARC and sequence number as in the update message (IE813: C_UPD_DAT) to former Consignee to inform him that the movement has changed destination.

Before the change of destination back to a former MSA of Destination it is possible that multiple actions have taken place. Changes of destination to other MSAs of Destinations may have been performed also. For simplicity it is assumed that no other (intermediate) changes of destination have occurred.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The Consignor initiates the change of destination process by submitting an update message (IE813: C_UPD_DAT) to change the MS of Destination and set the new MS of Destination as (one of) the former MS(s) of Destination. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it successfully. The MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

The MSA dispatch application includes the last sequence number of the ARC (SeqNo = PreviousSeqNo + 2) in the validated update message (IE813: C_UPD_DAT) and sends it to the former MSA destination application as well as to the Consignor as an acknowledgement.

In addition, the MSA dispatch application generates and sends to the new MSA of Destination (which was a former MSA of Destination) an e-AD (IE801: C_EAD_VAL) that includes the following information:

- The ARC of the update message (IE813: C_UPD_DAT), which is the same as in the original e-AD;
- The sequence number of the ARC which is incremented again by 1 (PreviousSeqNo + 2);
- The updated destination details (new Consignee, which was a former Consignee and Place of Delivery), as declared in the update message (IE813: C_UPD_DAT).

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

Upon the reception of the updated e-AD (IE801: C_EAD_VAL), the new MSA destination application (which was a former MSA of Destination) receives and validates the e-AD. Assuming that the validation passes successfully, the new MSA destination application sends the e-AD (IE801: C_EAD_VAL) to the new Consignee (which was a former Consignee) to inform him that he is the new Consignee of the movement. Moreover, the new MSA destination application sets the state of the movement to “Accepted”.

At the other side, the former MSA destination application validates the structure of the received update message (IE813: C_UPD_DAT). Assuming that the message structure validation passes successfully, the former MSA destination application sets the state of e-AD to “Diverted” and sends a notification message (IE803: C_EAD_NOT) with the same ARC and sequence number as in the update message (IE813: C_UPD_DAT) to the former Consignee to inform him that the movement has changed destination.

Although the state remains open (in “Diverted” state) at the former MSA of Destination, the latter can be acknowledged of the status at the MSA of Dispatch through the Manual Status Request/Response (i.e. Section III.VI.2.2). In case the Manual Status Response (IE905) to this request (IE904) proves that the e-AD at the MSA of Dispatch is in a final state, then the former MSA of Destination can drive this e-AD to the corresponding final state using its own means. It is the responsibility of the former MSA of Destination to ensure that the movement is not closed at the MSA of Dispatch, since it will still be possible to divert back to its territory.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The scenario to change the MS of Destination where the Consignor diverts a movement back to a (former) MSA of Destination is depicted in Figure 28 and Figure 29:

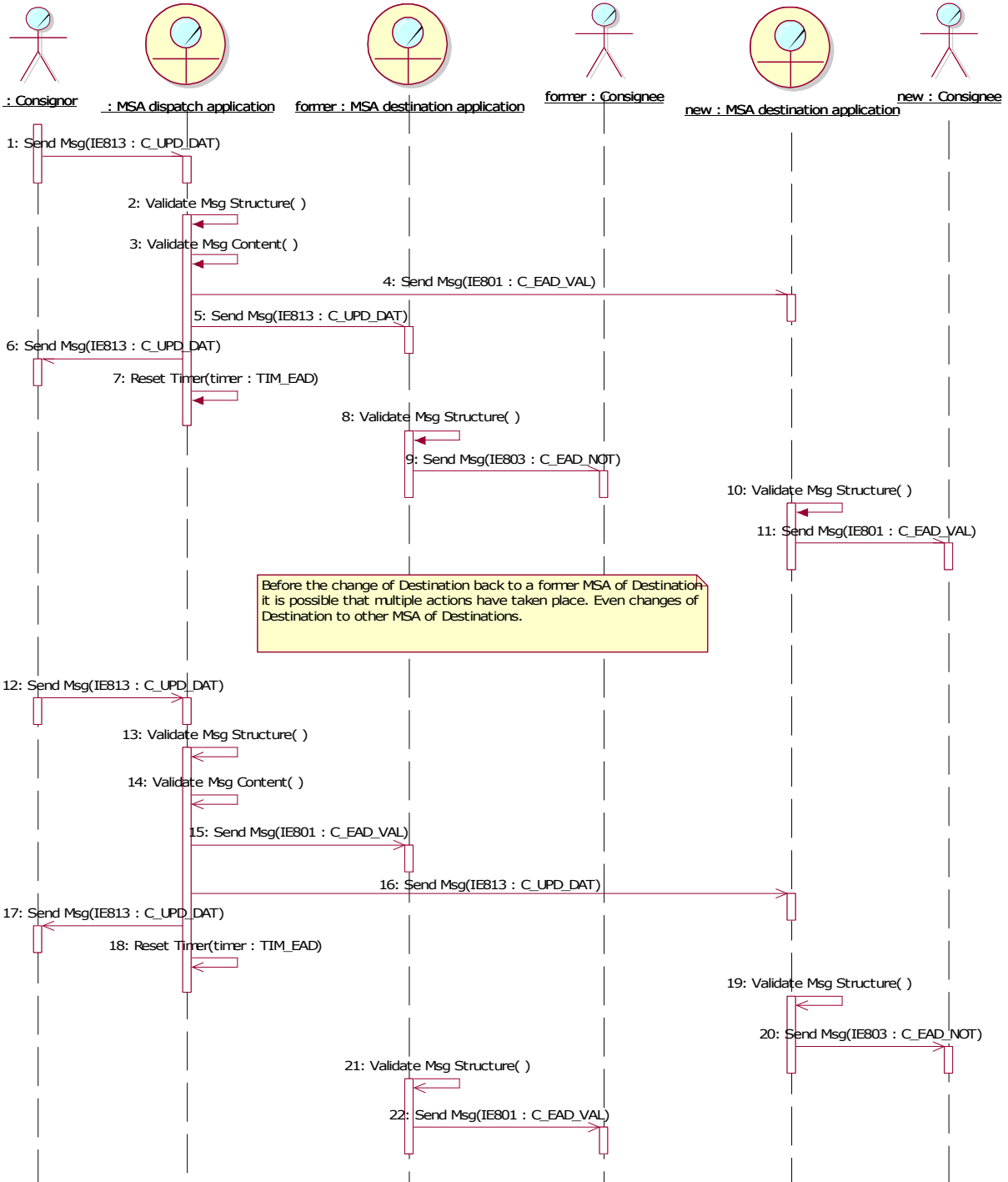


Figure 28: TSD - Change MS of Destination where the Consignor diverts a movement back to a (former) MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

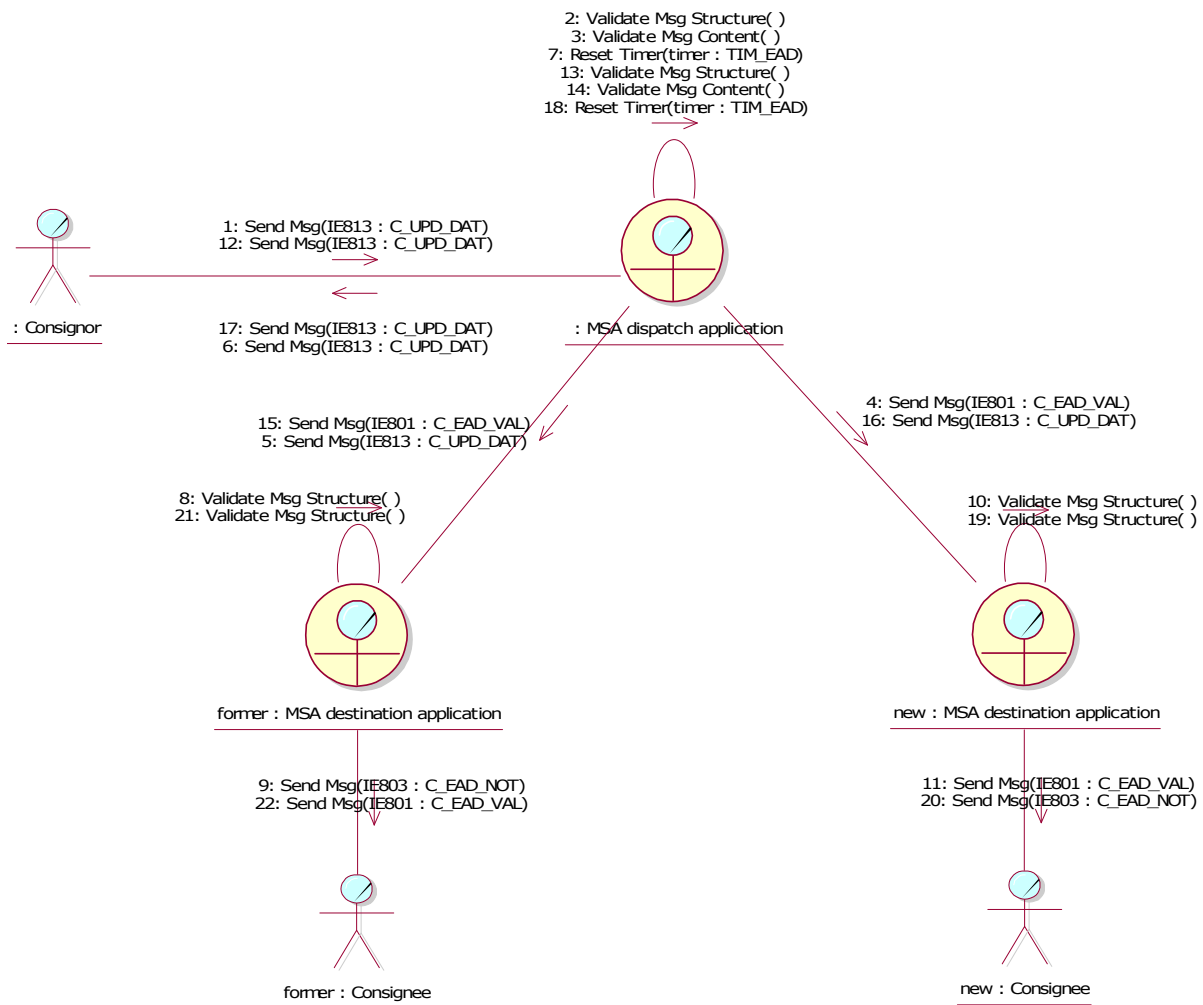


Figure 29: CLD - Change MS of Destination where the Consignor diverts a movement back to a (former) MSA of Destination

III.I.5.3 Return to the tax warehouse of dispatch

The Consignor initiates the change of destination process by submitting an update message (IE813: C_UPD_DAT) for return of goods thus changing the MS of Destination. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

The MSA dispatch application includes the last sequence number of the ARC incremented by 1 (PreviousSeqNo + 1) in the validated update message (IE813: C_UPD_DAT) and sends it to the MSA destination application as well as to the Consignor as an acknowledgement.

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon the reception of the update message (IE813: C_UPD_DAT), the MSA destination application validates the structure of the received message. Assuming that the message structure validation passes successfully, the MSA destination application sets the state of e-AD to “*Diverted*” and sends a notification message (IE803: C_EAD_NOT) with the same ARC and sequence number as in the update message (IE813: C_UPD_DAT) to the Consignee to inform him that the movement has changed destination.

To complete the return to the tax warehouse of dispatch and discharge the e-AD, the Consignor submits a draft Report of Receipt (IE818: C_DEL_DAT) indicating delivery acceptance to the MSA dispatch application. The message structure and content are validated by the MSA dispatch application that complete successfully. Then the MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “*Delivered*”, which is a final state. Finally, the MSA dispatch application sends back the validated RoR to the Consignor as a confirmation.

The scenario to change return to the tax warehouse of dispatch is depicted in Figure 30 and Figure 31:

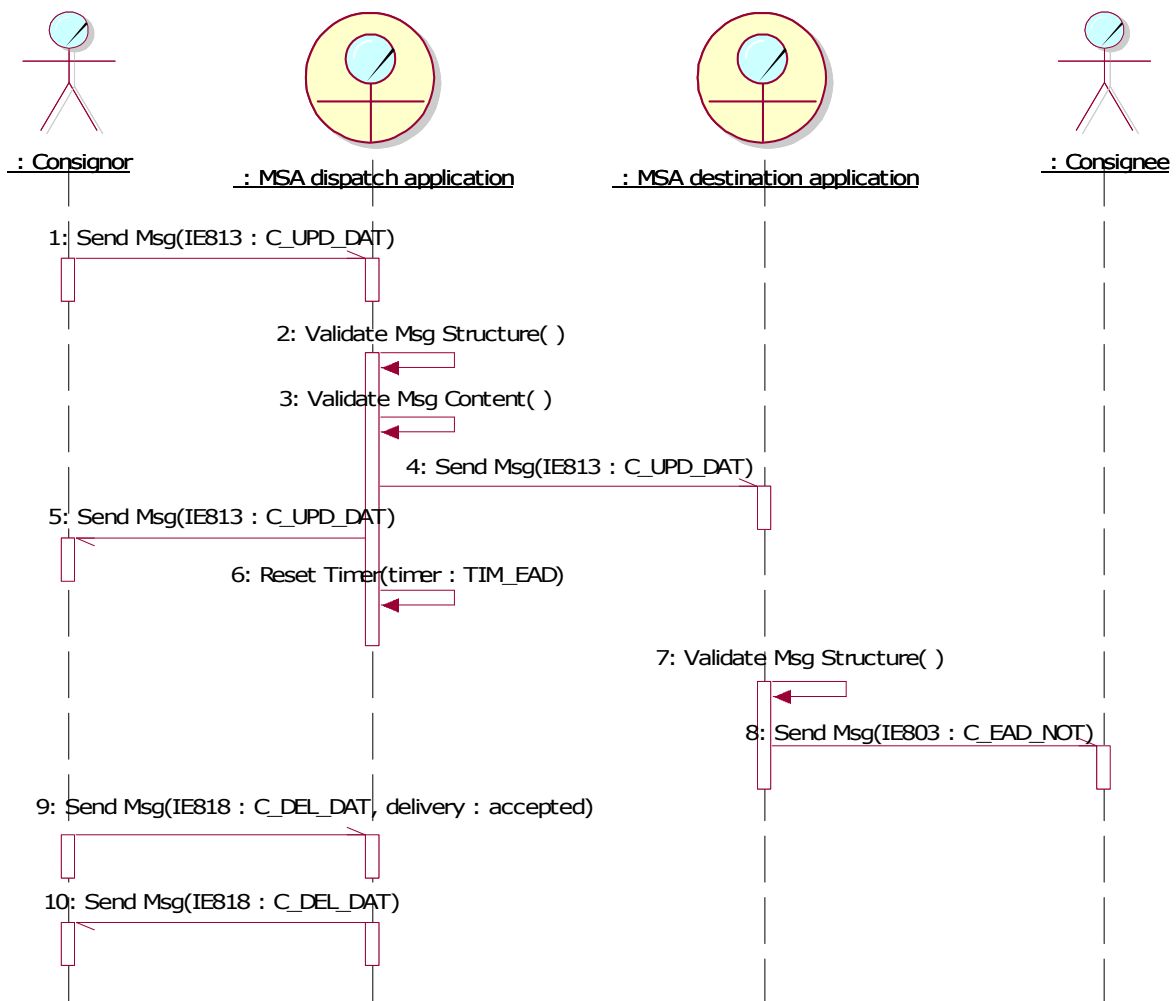


Figure 30: TSD - Return to the tax warehouse of dispatch

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

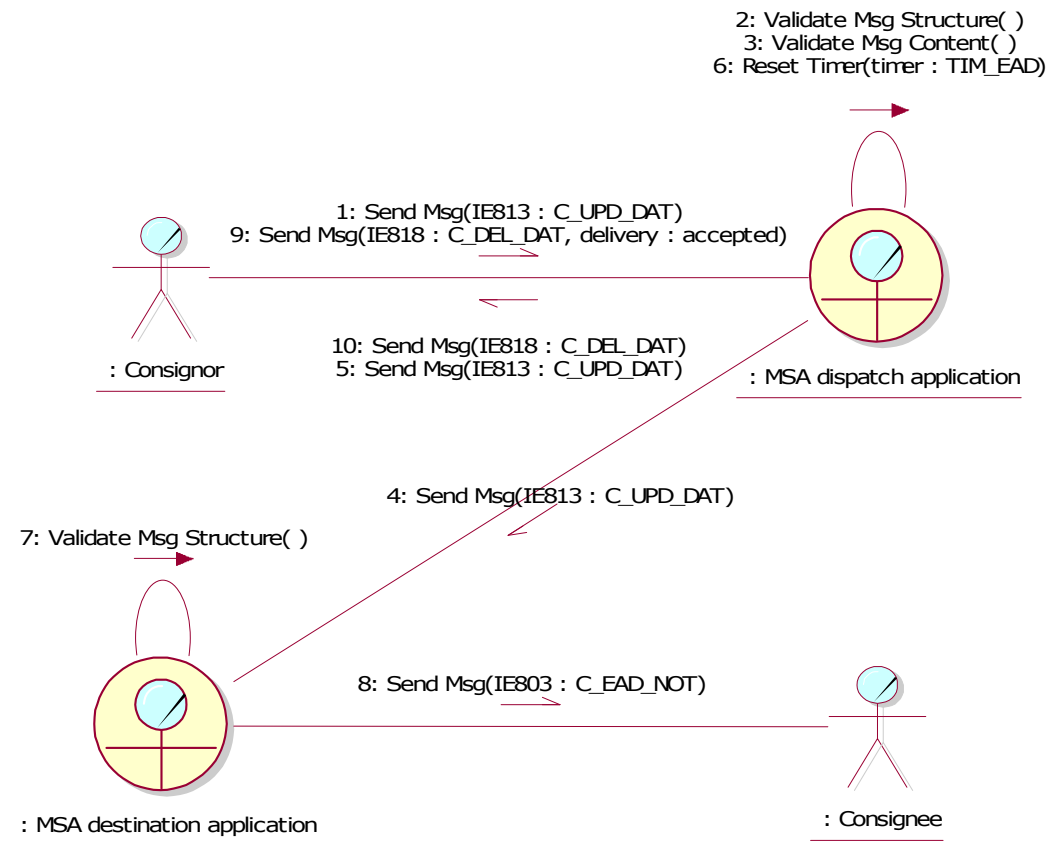


Figure 31: CLD - Return to the tax warehouse of dispatch

III.I.5.4 Change of Consignee (not the MS of Destination)

The Consignor initiates the change of destination process to change the Consignee (hence, the Place of Delivery as well) and not the MS of Destination. In that case, the Consignor sends the draft update (IE813: C_UPD_DAT) for validation to the MSA dispatch application. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). Assuming that the draft update is valid, the MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

The MSA dispatch application includes the last sequence number of the ARC incremented by 1 (PreviousSeqNo + 1) in the validated update message (IE813: C_UPD_DAT) and sends it to the Consignor as an acknowledgement.

In addition, the MSA dispatch application sends to the unchanged MSA of Destination an e-AD (IE801: C_EAD_VAL) that includes the following information:

- The ARC of the update message (IE813: C_UPD_DAT)¹;
- The sequence number of the ARC which is incremented by 1 (PreviousSeqNo + 1);

¹ Hence, the ARC is the same as in the original e-AD

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The updated destination details (new Consignee and Place of Delivery), as declared in the update message (IE813: C_UPD_DAT).

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

Upon the reception of the e-AD (IE801: C_EAD_VAL), the MSA destination application validates the structure of the received e-AD (IE801: C_EAD_VAL). In addition, the MSA destination application checks whether the same e-AD (IE801: C_EAD_VAL) instance (same ARC and same “(HEADER) E-AD.Sequence Number”) has already been received. Assuming that the message structure validation passes successfully and that the e-AD (IE801: C_EAD_VAL) instance is unique (it contains a unique “(HEADER) E-AD.Sequence Number”), the MSA destination application accepts and processes the message. The MSA destination application changes the state of the e-AD at MSA of Destination to “Accepted”.

Finally, the MSA destination application sends:

- A notification message (IE803: C_EAD_NOT) with the same sequence number as in the update message (IE813: C_UPD_DAT) to the former Consignee to inform him that the consignment has changed destination;
- The e-AD (IE801: C_EAD_VAL) to the new Consignee to notify him that he is the new Consignee of the consignment.

The scenario to change the Consignee but not the MS of Destination is depicted in Figure 32 and Figure 33:

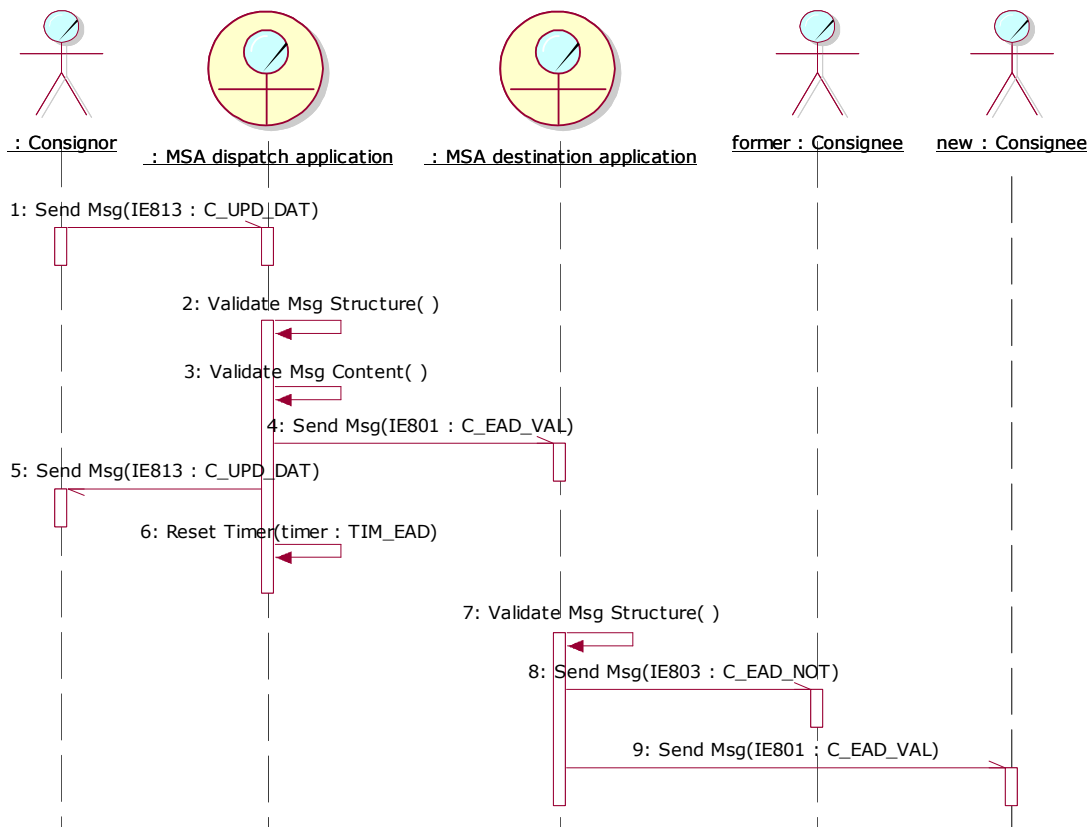


Figure 32: TSD - Change of Consignee following the Submission of e-AD (MS of Destination unchanged)

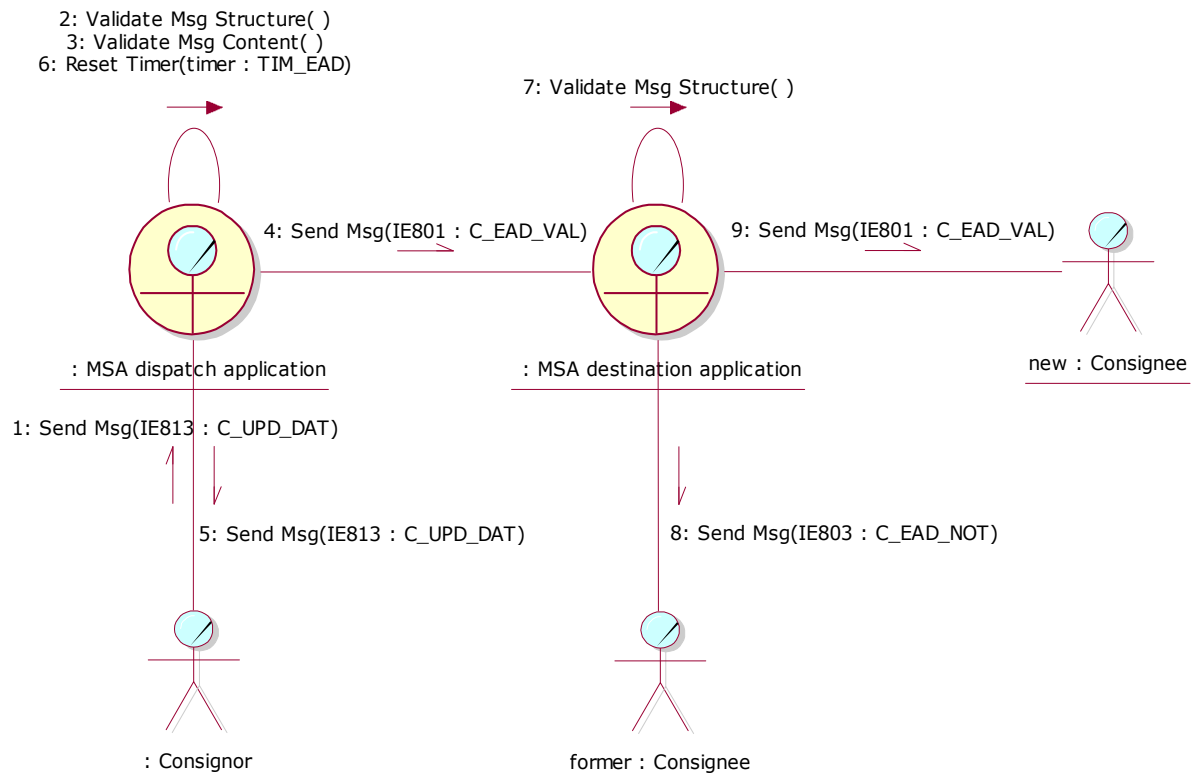


Figure 33: CLD - Change of Consignee following the Submission of e-AD (MS of Destination unchanged)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.5.5 Change of Place of Delivery

The Consignor initiates the change of destination process to change the Place of Delivery. In that case, the Consignor sends the draft update (IE813: C_UPD_DAT) for validation to the MSA dispatch application. The MSA dispatch application receives the draft update message (IE813: C_UPD_DAT) and validates it successfully (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI). The MSA dispatch application changes the state of the e-AD at MSA of Dispatch to “Accepted”.

The MSA dispatch application includes the last sequence number of the ARC incremented by 1 (PreviousSeqNo + 1) in the validated update message (IE813: C_UPD_DAT) and submits it to the MSA destination application and to the Consignor as acknowledgement.

Finally, in case the journey time has changed, the MSA dispatch application updates the TIM_EAD timer if the expected end of the movement is still in the future or first resets the flag that has been raised locally at expiration time and then restarts the timer if the expected end of the movement is in the past.

Upon the reception of the update message (IE813: C_UPD_DAT), the MSA destination application validates the structure of the received message. The MSA destination application changes the state of the e-AD at MSA of Destination to “Accepted” and forwards the update message (IE813: C_UPD_DAT) to the Consignee.

The scenario to change the Place of Delivery is depicted in Figure 34 and Figure 35:

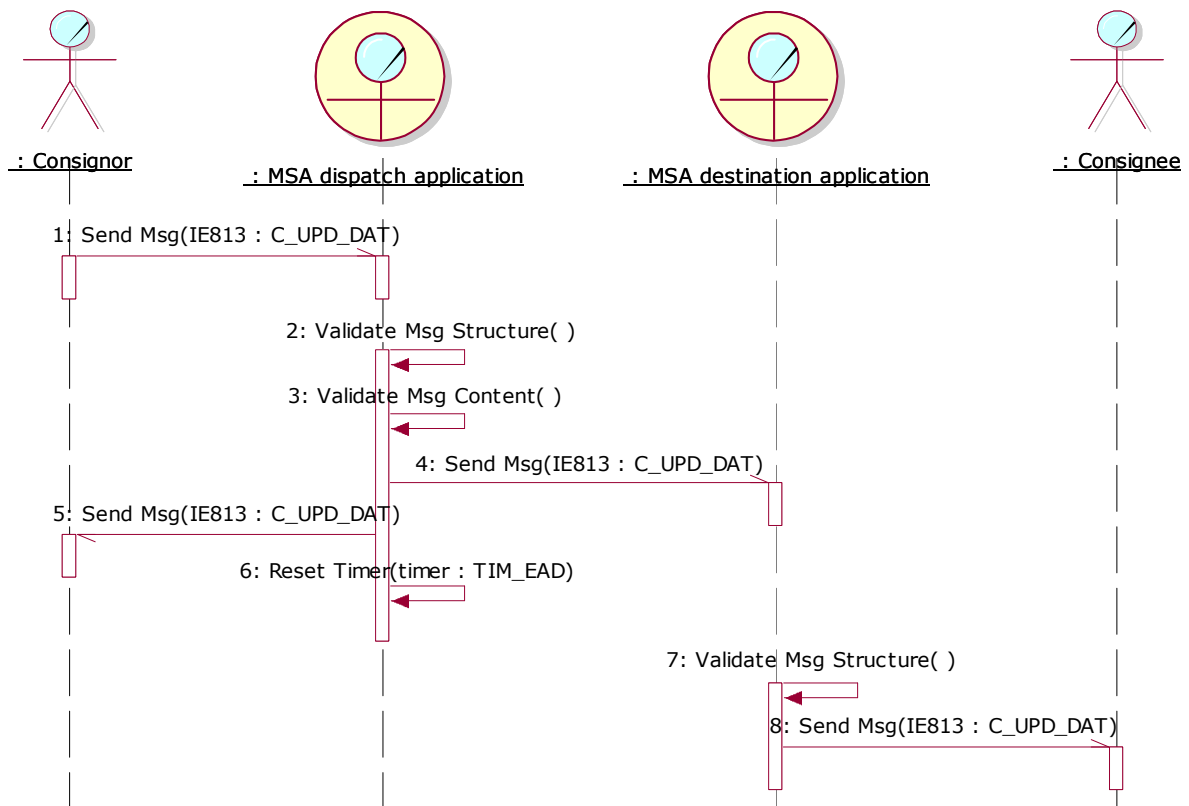


Figure 34: TSD - Change of Place of Delivery following the Submission of e-AD (Consignee unchanged)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

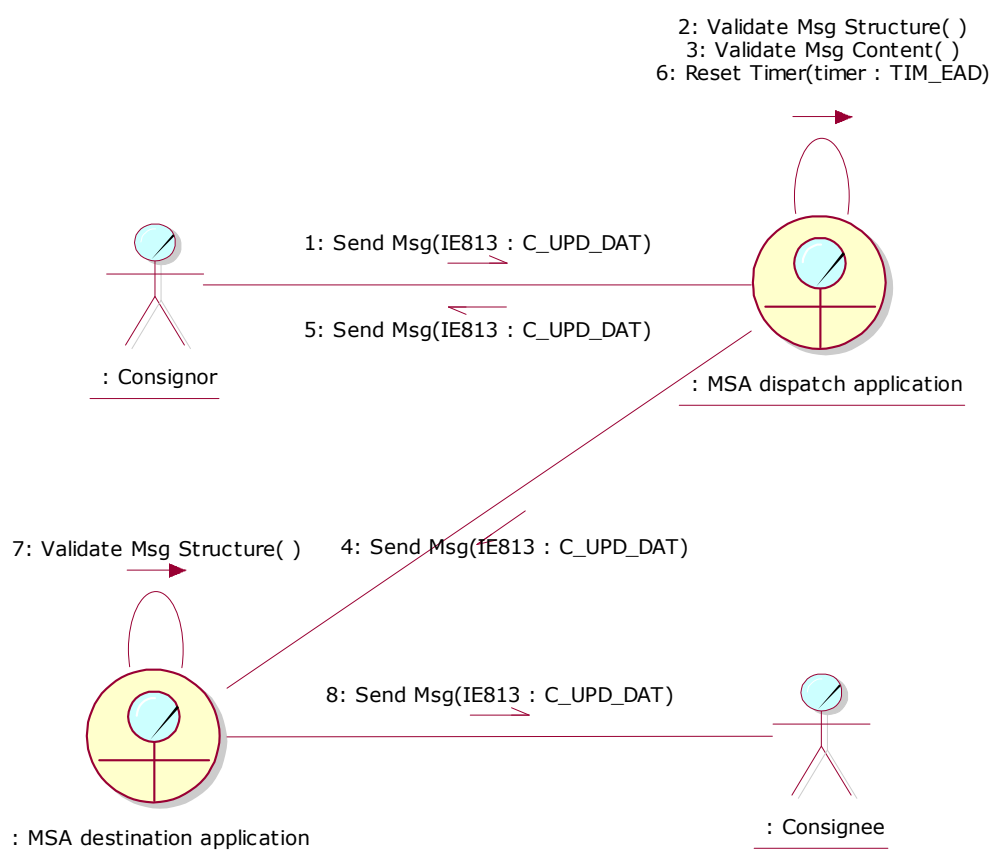


Figure 35: CLD - Change of Place of Delivery following the Submission of e-AD (Consignee unchanged)

III.I.6 Splitting of Consignment (UC2.36)

Consignments of energy products may be split into two or more parts thus replacing the original consignment. The successful completion of a splitting operation causes state transition for the concerned e-AD to “*Replaced*” pertaining to both the MSA of Dispatch and the MSA of Destination; the new e-ADs resulting from the splitting operation are registered at the “*Accepted*” state at the MSA of Destination and all involved MSAs.

The new destination can be any type other than exempted Consignee namely a tax warehouse or the premises of a registered Consignee or of a temporary registered Consignee or a place of direct delivery or export or unknown destination. At most one of the new destinations is the same as for the initial e-AD including the case where the initial destination was unknown. The splitting operation must not change the goods or their quantities.

The scenario of this use case requires the state of the concerned e-AD to be in the “*Accepted*”, “*Refused*”, “*Partially refused*”, or “*Rejected*” state. Moreover, the scenario of this use case requires the state of the concerned e-AD to be in the “*Exporting*” state in which case the scenarios for the exportation of goods apply described in Sub-Section III.II Export Scenarios.

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.II.1.2.3	Change of Destination (UC2.05)	Accepted

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.6.2	Change of Destination (UC2.05)	Accepted
III.II.2.8	Export Operation at Office of Export and movement not released by Customs followed by change of destination	Accepted
III.II.3.3.2	Change of Destination (UC2.05)	Accepted
III.II.3.6.2	Change of Destination (UC2.05)	Accepted
III.I.2.2	e-AD rejected	Rejected
III.I.4.2	Delivery Refused	Refused
III.I.4.3	Delivery Partially Refused	Partially refused

Table 12: Scenarios that must have been previously completed in order to submit a splitting of consignment

The splitting operation may differ depending on the types of new destinations and consignees. The following scenario satisfies all cases by instantiating the MSA destination application and the Consignee actors. The two instances of the aforementioned actors, namely the *former* and the *new*, should be considered to apply to different types of new destinations. There may be cases where the *former MSA destination* and *new MSA destination* are the same MSA². The scenario discussion that follows considers these different cases.

The basic scenario of the splitting operation begins as soon as the Consignor submits a draft splitting operation (IE825: E_SUPL_SUB) to the MSA dispatch application concerning an e-AD in the “Accepted”, “Refused”, “Partially refused”, “Rejected”, or “Exporting” state. The Consignor is waiting for a notification message (IE803: C_EAD_NOT) to be received from the MSA dispatch application as well as the new e-ADs (IE801: C_EAD_VAL).

The MSA dispatch application receives the draft splitting operation (IE825: E_SUPL_SUB). Upon successful validation of the incoming message (the exceptional cases of the message validations in this scenario are described in Sub-Section III.VI), the MSA dispatch application updates the state of the e-AD to “Replaced” and forwards a notification message (IE803: C_EAD_NOT) to the former MSA destination application and back to the Consignor. The sequence number of the ARC in the notification message (IE803: C_EAD_NOT) is the last sequence number for the replaced e-AD. In case the destination of the initial e-AD is unknown, the MSA dispatch application will not forward a notification message (IE803: C_EAD_NOT) to the former MSA destination application.

If any timers associated with the e-AD to be split have already expired at the limit date, the MSA dispatch application resets their flag that has been raised locally at expiration time. If any timers associated with the e-AD to be split are still running, the MSA dispatch application stops them.

² Contrary to the change of destination scenarios (see III.I.5) where the former MSA destination and new MSA destination are always different MSAs.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The MSA dispatch application creates new e-ADs according to the information of the draft splitting operation (IE825: E_SUPL_SUB). For each of the created e-ADs the MSA dispatch application assigns an ARC to the e-ADs and creates a validated e-AD (IE801: C_EAD_VAL). The state of each new e-AD at the MSA dispatch application is set to “*Accepted*”. In addition, for each new e-AD the timer TIM_EAD is initiated to expire at the expected end of movement (i.e. date of dispatch plus journey time). At most one of the created e-ADs may have unknown destination in which case the MSA dispatch application initiates the timer TIM_FDF to expire at the limit date for filling the destination fields.

If one of the created e-ADs contains the same destination information as the initial e-AD and the destination is not unknown, then the MSA dispatch application forwards the created e-AD (IE801: C_EAD_VAL) to the former MSA destination application. For each of the created e-ADs having different destination information as the initial e-AD and having the destination not unknown the MSA dispatch application forwards the created e-AD (IE801: C_EAD_VAL) to the new MSA destination applications (i.e. one for each new MSA destination application).

If the destination of the initial e-AD is not unknown, then the former MSA destination application has received a notification message (IE803: C_EAD_NOT) from the MSA dispatch application. In that case the MSA destination application validates the notification message (IE803: C_EAD_NOT). Upon successful validation of the incoming message, the MSA destination application changes the state of the e-AD to “*Replaced*” and forwards the cancellation notification (IE803: C_EAD_NOT) to the former Consignee.

If one of the created e-ADs contains the same destination information as the initial e-AD and the destination is not unknown, then the former MSA destination application has received a created e-AD (IE801: C_EAD_VAL) from the MSA dispatch application. In that case the MSA destination application validates the created e-AD (IE801: C_EAD_VAL). Upon successful validation of the incoming message, the MSA destination application registers the e-AD as “*Accepted*” and forwards the e-AD (IE801: C_EAD_VAL) to the former Consignee.

For each of the created e-ADs having different destination information as the initial e-AD and having the destination not unknown there is a *new* MSA destination application instance. The scenario description shall henceforth refer to the *new* MSA destination application actor for each of these instances. Note that the actions performed by the *new* MSA destination application are essentially the same with the action performed after the reception of e-AD (IE801: C_EAD_VAL) as in the scenarios of Section III.I.1 Submission and registration of an e-AD (UC2.01).

Upon the reception of the created e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the *new* MSA destination application stores the e-AD and sets the state of the e-AD at MSA of Destination to “*Accepted*”. Finally, the MSA destination application forwards the e-AD (IE801: C_EAD_VAL) to the Consignee.

The scenario to split a consignment is depicted in Figure 36 and Figure 37:

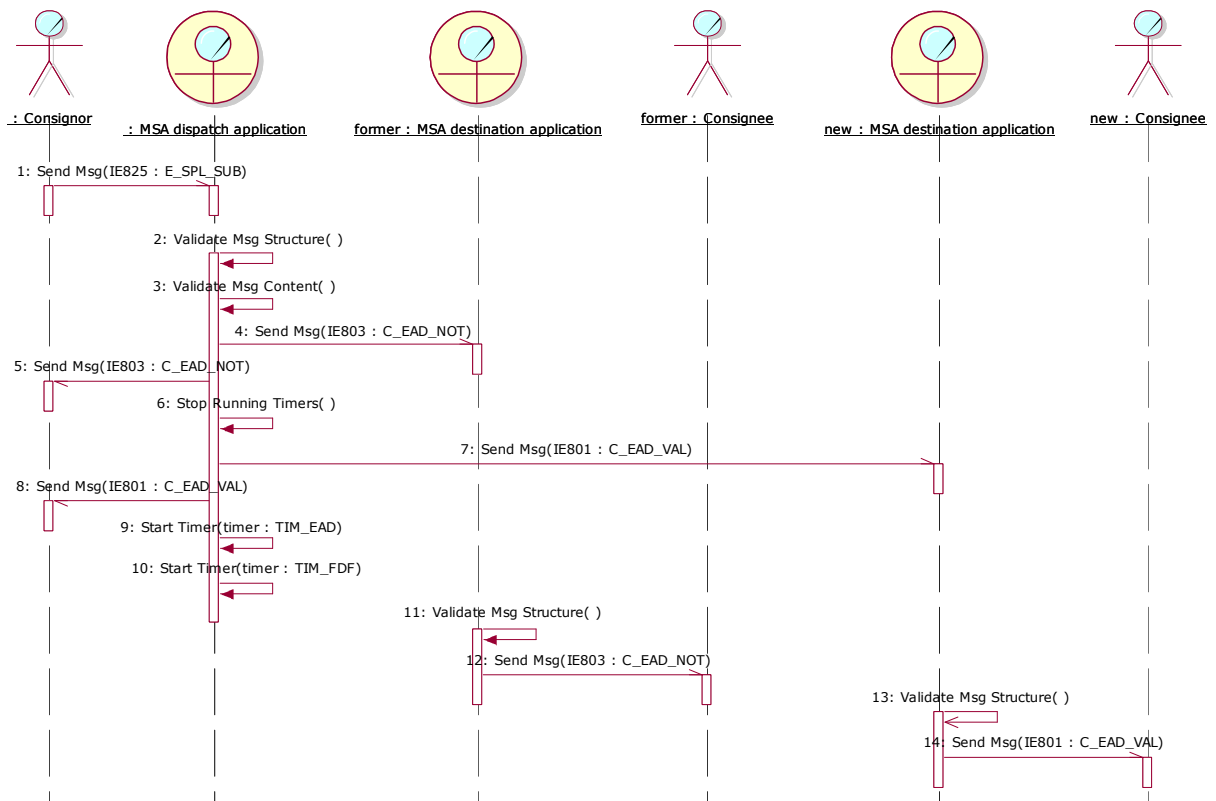


Figure 36: TSD - Splitting of Consignment (UC2.36)

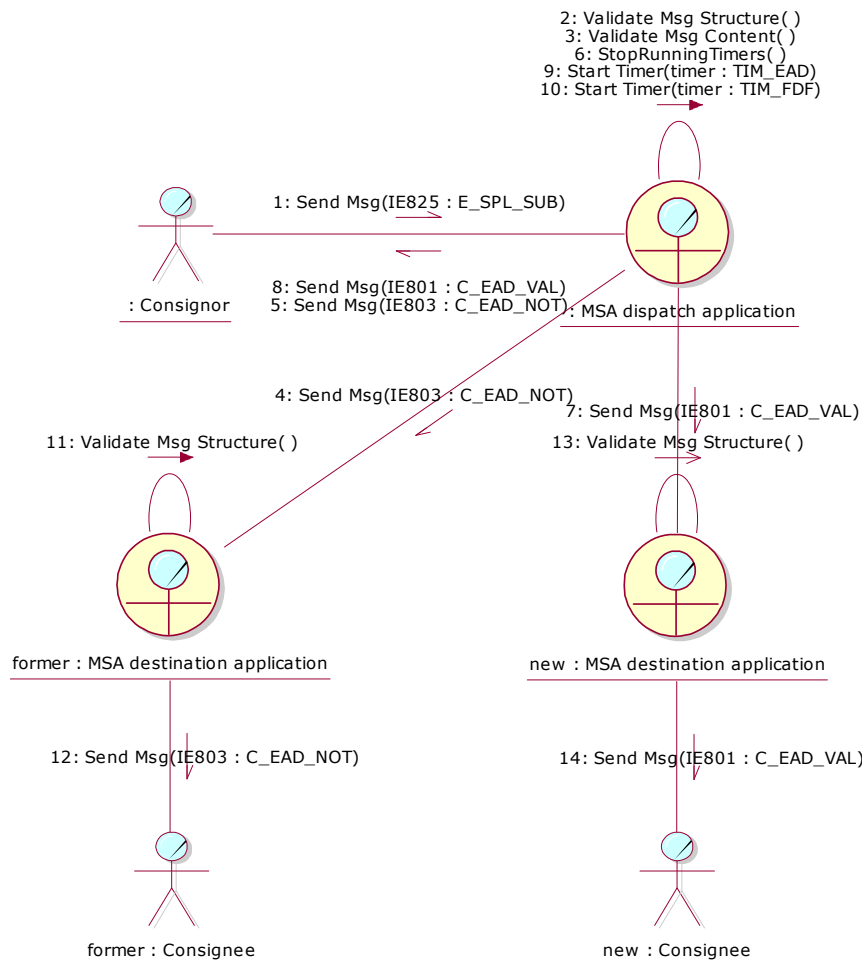


Figure 37: CLD - Splitting of Consignment (UC2.36)

III.I.7 Reminder at expiry of time limit for Report of Receipt (UC2.33)

The MSA dispatch application upon receiving the valid draft e-AD initiates the timer TIM_EAD to expire at the expected end of movement (see III.I.1 Submission and registration of an e-AD (UC2.01)). It is expected from the Consignee to acknowledge the receipt of goods by submitting a Report of Receipt (RoR) (see III.I.4 Submission of Report of Receipt (UC2.06)) before the TIM_EAD timer expiration. This use case describes the message exchange protocol when TIM_EAD timer expires before the Consignee has sent a RoR.

The scenarios of this use case require the state of the concerned e-AD to be in the “Accepted” state.

Section	Scenario	Ending State
III.I.1.1.1	Origin is tax warehouse and the destination is known	Accepted
III.I.1.2.2	Origin is import and the destination is known	Accepted
III.II.1.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.2.3	Change of Destination (UC2.05)	Accepted
III.II.2.6.2	Change of Destination (UC2.05)	Accepted
III.II.2.8	Export Operation at Office of Export and movement not released by Customs followed by change of destination	Accepted
III.II.3.3.2	Change of Destination (UC2.05)	Accepted

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.3.6.2	Change of Destination (UC2.05)	Accepted
--------------	--------------------------------	----------

Table 13: Scenarios that must have been previously completed in order to submit a reminder at expiry of time limit for Report of Receipt

This use case is described with two sequential scenarios. The first scenario (section III.I.7.1) describes the message exchange protocol after the TIM_EAD timer expiration at the MSA dispatch application and must be completed. The second scenario (section III.I.7.2) describes the submission of explanations on delay for delivery and is optional.

III.I.7.1 Status Request/Response after the TIM_EAD timer expiration

The scenario described includes the triggering of automatic Status Request/Response by the MSA dispatch application to examine why the TIM_EAD timer expired without receiving the RoR. This mechanism enables the MSA dispatch application to identify when the RoR has not been received due to technical problems and when the RoR has not been sent by the Consignee within the allocated time.

This scenario assumes that all message validations pass successfully; the exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

When the TIM_EAD timer expires, the MSA dispatch application sends a Status Request (IE904: C_STD_REQ) to the MSA destination application for the specific ARC by including the sequence number of the last business (event) message sent to the MSA of Destination and the information that the e-AD at MSA of Dispatch is found on the “*Accepted*” state. The last message received from the MSA destination application may be none or an alert or rejection (IE819: C_REJ_DAT).

The MSA destination application receives the Status Request (IE904: C_STD_REQ) and examines the contained information. The MSA destination application responds to that request with the Status Response (IE905: C_STD_RSP) message mentioning the last known sequence number of the movement and that the MSA destination application is found on the “*Accepted*” state.

It may be the case that the state of the MSA destination application has sent the Report of Receipt, but the MSA dispatch application has not received the RoR message. In this case the Status Response (IE905: C_STD_RSP) sent by the MSA destination application will contain either the “*Delivered*”, the “*Refused*” or the “*Partially Refused*” state. This case is described in the TIM_EAD timer expiration after Missed RoR scenario of Section III.VI.2.4.2.

Upon the reception and successful validation of the Status Response (IE905: C_STD_RSP) from the MSA destination application, the MSA dispatch application understands that the Consignee has not sent the RoR within the allocated time. Hence, the MSA dispatch application automatically flags the concerned e-AD to allow further retrieval for examination by verification officers and generates a reminder/flagging message (IE802: C_EXC_REM) that is sent to the Consignor and to the MSA destination application as a notification of the RoR delay.

Upon reception and successful validation of the flagging message (IE802: C_EXC_REM), the MSA destination application automatically flags the concerned e-AD to allow further retrieval for examination by verification officers and forwards it (optionally), if relevant, to the Consignee to provide his/her own explanations on the delay.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

At this point of the scenario, the Consignor and optionally the Consignee have been notified that the arrival of a consignment is late. The scenario may optionally continue with the submission of explanations on delay for delivery by either the Consignor or the Consignee, which is described in the next section (see III.I.7.2). Alternatively, the scenario may end here:

- if the Consignor submits a late e-AD cancellation provided that the consignment has not left the Place of Dispatch, as described in section III.I.3 Cancellation of an e-AD by the Consignor (UC2.10);
- if the Consignee sends the expected RoR, as described in section III.I.4 Submission of Report of Receipt (UC2.06);
- if the Consignor submits a change of destination, as described in section III.I.5 Change of Destination (UC2.05);
- if the Consignor splits the consignment, provided that the consignment contains energy products as described in section III.I.5 Change of Destination (UC2.05).

The scenario for sending reminder message after expiry of the time limit for Report of Receipt thus far is depicted in Figure 38 and Figure 39. No state transitions result from the completion of the scenario.

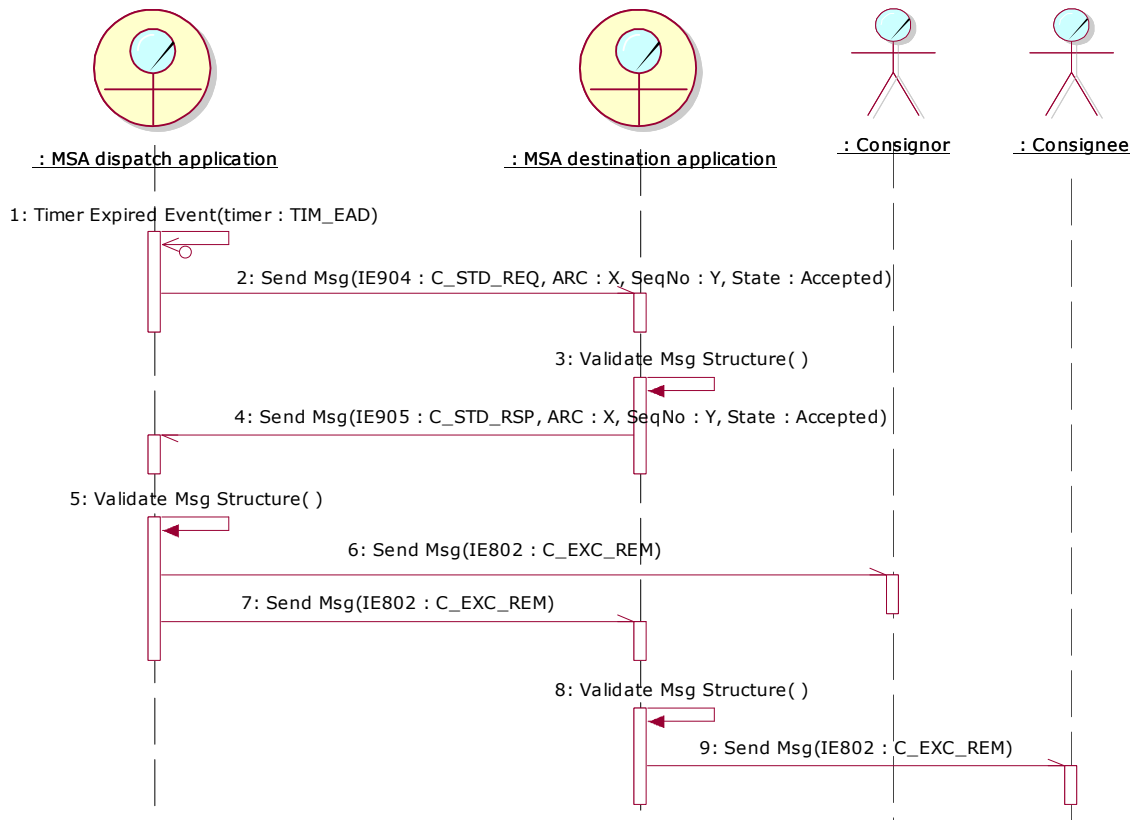


Figure 38: TSD - Status Request/Response after the TIM_EAD timer expiration - Reminder for RoR

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

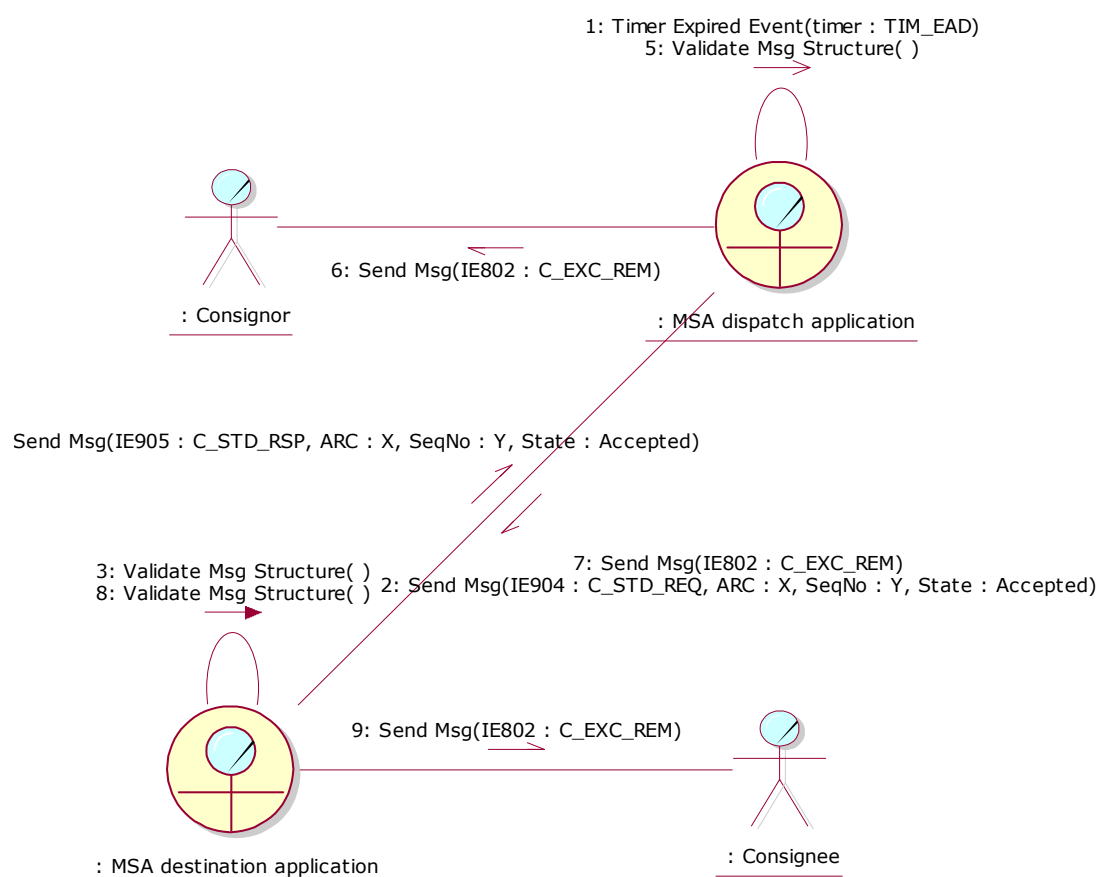


Figure 39: CLD - Status Request/Response after the TIM_EAD timer expiration - Reminder for RoR

III.I.7.2 Submission of explanations on delay for delivery

This scenario is complementing the previous scenario for sending reminder message after expiry of the time limit for Report of Receipt (see III.I.7.1). According to the previous scenario the Consignor and optionally the Consignee have been notified that the arrival of a consignment is late.

This scenario assumes that all message validations pass successfully; the exceptional cases of the aforementioned validations are described in Sub-Section III.VI.

The Consignor and/or the Consignee may optionally submit an explanation message (IE837:C_DEL_EXP) to the MSA dispatch or destination application, respectively. Both cases are covered with one scenario using actor instantiations. The “*Consignor / Consignee*” instance of the *Economic operator* actor should be considered as either the *Consignor* or the *Consignee* actor. The “*dispatch / destination*” and the “*destination / dispatch*” instance of the *MSA application* actor should be considered as either the *MSA dispatch application* or the *MSA destination application* actor.

If the scenario is initiated by the *Consignor*, then “*dispatch / destination : MSA application*” should be considered as the *MSA dispatch application* and the “*destination / dispatch : MSA application*” should be considered as the *MSA destination application*. If the scenario is initiated by the *Consignee*, then “*dispatch / destination : MSA application*” should be considered as the *MSA destination application* and the “*destination / dispatch : MSA application*” should be considered as the *MSA dispatch application*.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The *Consignor / Consignee* Economic operator submits explanations on delay for delivery (IE837: C_DEL_EXP) to the *dispatch / destination MSA application* for a consignment having late delivery.

The *dispatch / destination MSA application* receives the explanations on delay for delivery (IE837: C_DEL_EXP) and performs validation. Upon successful validation, the *dispatch / destination MSA application* forwards the incoming explanations on delay for delivery (IE837: C_DEL_EXP) message to the *destination / dispatch MSA application*.

The *destination / dispatch MSA application* validates the message structure of the received explanations on delay for delivery (IE837: C_DEL_EXP).

It is to be noted that when the Consignee has initiated this scenario the provision of explanations does not relieve the Consignee from his/her obligation to submit the RoR as soon as possible.

The scenario for the submission of explanations on delay for delivery is depicted in Figure 40 and Figure 41. No state transitions result from the completion of the scenario.

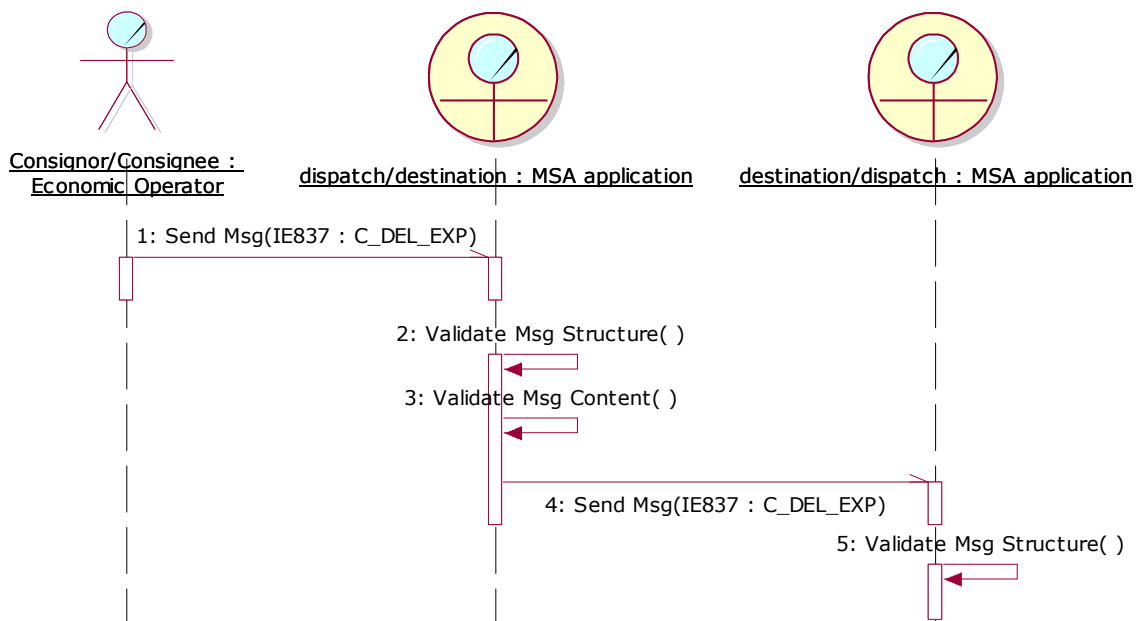


Figure 40: TSD - Submission of explanations on delay for delivery - Reminder for RoR

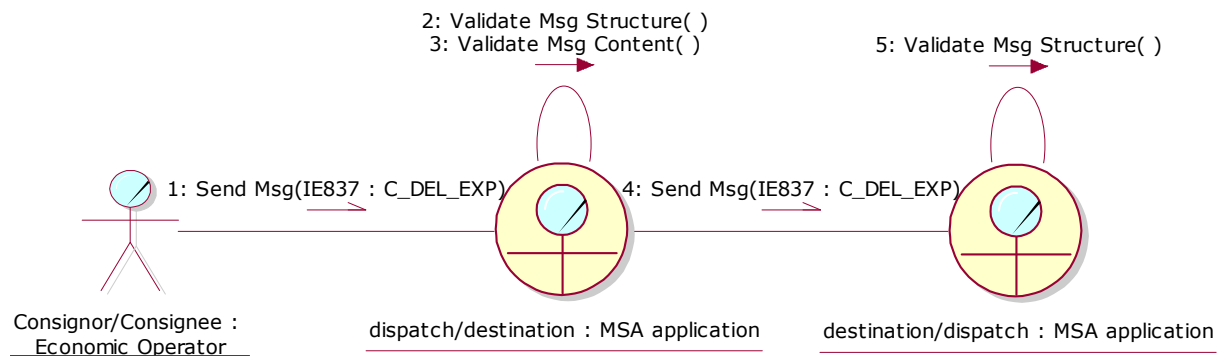


Figure 41: CLD - Submission of explanations on delay for delivery - Reminder for RoR

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.8 Reminder at expiry time for change of destination (UC2.17)

For the cases when the delivery of a consignment has been refused or partially refused (as described in III.I.4.2 and III.I.4.3, respectively) or the e-AD has been rejected (as described in III.I.2.2) by the Consignee, then the MSA dispatch application has already initiated the TIM_CHS timer. In all of the three aforementioned cases the Consignor is expected to either:

- initiate the change of destination process as described in Chapter III.I.5; or
- initiate the splitting of consignment process as described in Chapter III.I.6; or
- initiate the cancellation of an e-AD process as described in Chapter III.I.3.

If any of the aforementioned processes is performed by the Consignor, the MSA dispatch application will reset the TIM_CHS timer or stop it as described in the corresponding scenarios. If, however, none of the aforementioned processes is performed by the Consignor, the TIM_CHS timer will expire at the MSA dispatch application, which is the opening action of this use case and consequently of this scenario.

The MSA dispatch application flags the e-AD so that it can set the timer when expired. The implementation mechanism of the flag depends on the national system to be developed. Finally, the MSA dispatch application sends a reminder (IE802: C_EXC_REM) to the Consignor. This reminder will include the ARC of the movement that has been refused or partially refused, the identity of the declared Consignee and the limit date that the Consignor should respond.

The scenario for sending reminder message after expiry of the change of destination timer is depicted in Figure 42 and Figure 43:

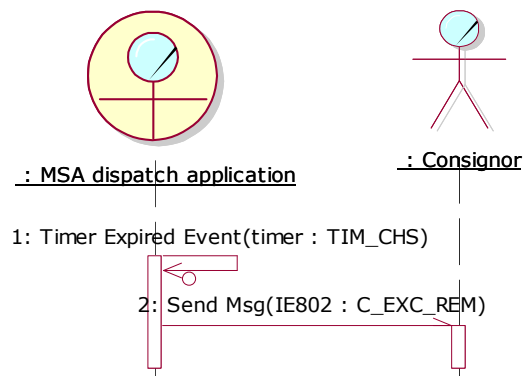


Figure 42: TSD - Sending reminder message after expiry of the change of destination timer

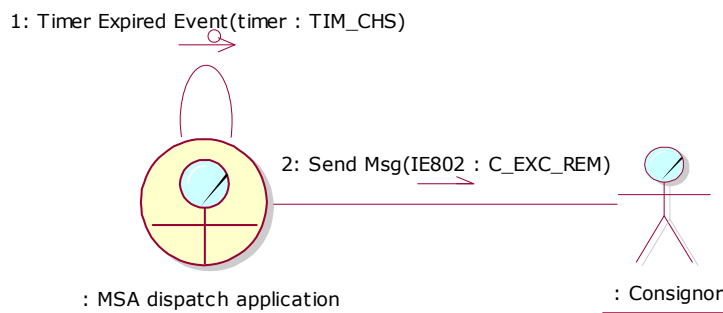


Figure 43: CLD - Sending reminder message after expiry of the change of destination timer

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.9 Reminder at expiry of time limit for update of destination fields (UC2.34)

For the cases when the destination of a consignment is unknown (as described in III.I.1.1.2, III.I.1.2.3, and III.I.6) the MSA dispatch application has already initiated the TIM_FDF timer and the Consignor is expected to either:

- initiate the change of destination process as described in Chapter III.I.5; or
- initiate the splitting of consignment process as described in Chapter III.I.6; or
- initiate the cancellation of an e-AD process as described in Chapter III.I.3 in case the goods have not left the Place of Dispatch.

If any of the aforementioned processes is performed by the Consignor, the MSA dispatch application will reset the TIM_FDF timer or stop it as described in the corresponding scenarios. If, however, none of the aforementioned processes is performed by the Consignor, the TIM_FDF timer will expire at the MSA dispatch application, which is the opening action of this use case.

III.I.9.1 Expiration of the time limit to update the destination fields

After the TIM_FDF timer expiration, the MSA dispatch application sends a reminder (IE802: C_EXC_REM) to the Consignor. This reminder will include the ARC of the consignment with unknown destination and the limit date that the Consignor should respond.

The Consignor's response may be the change of destination, the splitting or the cancellation of the movement. In this case the scenario ends here and is complemented by the scenarios corresponding to the Consignor's response. Alternatively, the Consignor may send an explanations message with the reasons not to update the e-AD (this scenario is described in section III.I.9.2 below).

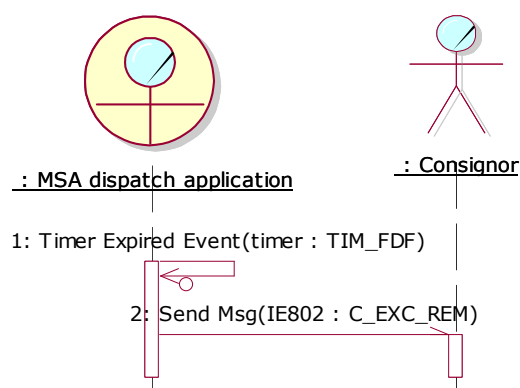


Figure 44: TSD - Sending reminder message after expiration of the time limit to update the destination fields

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

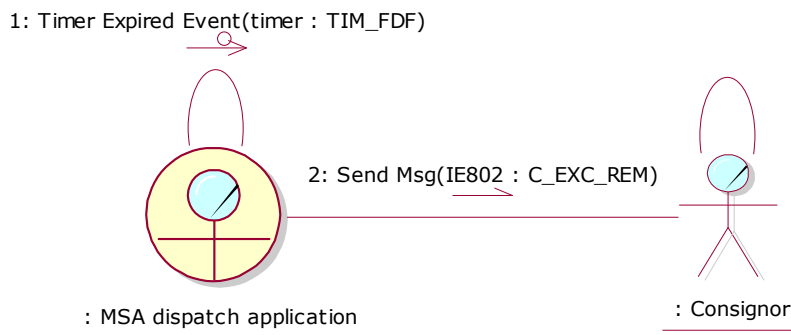


Figure 45: CLD - Sending reminder message after expiration of the time limit to update the destination fields

III.I.9.2 Expiration of the time limit to update the destination fields with Consignor explanations

After the TIM_FDF timer expiration, the MSA dispatch application sends a reminder (IE802: C_EXC_REM) to the Consignor. This reminder will include the ARC of the consignment with unknown destination and the limit date that the Consignor should respond.

The Consignor may send an explanations message (IE837: C_DEL_EXP) with the reasons not to update the e-AD to the MSA dispatch application. The MSA dispatch application receives the explanations message (IE837: C_DEL_EXP) and performs validation. Upon successful validation of the incoming message, the MSA dispatch application makes the message information available for examination by verification officers.

The exceptional cases of the unsuccessful validation of the explanations message received by the MSA dispatch application are described in Sub-Section III.VI.

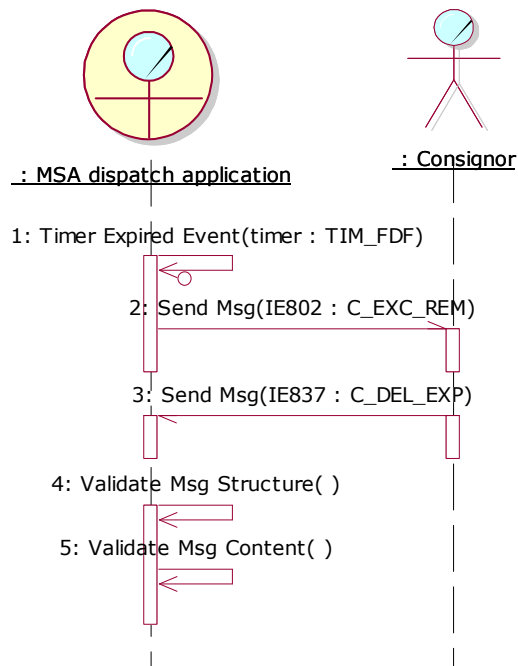


Figure 46: TSD - Sending reminder message after expiration of the time limit to update the destination fields with Consignor explanations

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

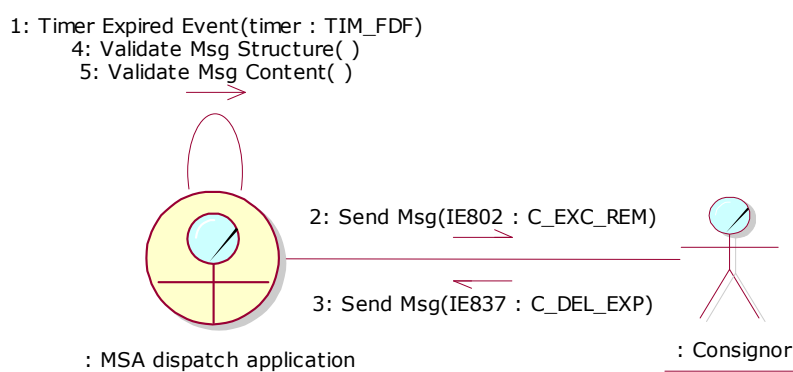


Figure 47: CLD - Sending reminder message after expiration of the time limit to update the destination fields with Consignor explanations

III.I.10 Post-delivery processing (UC2.12)

When a consignment reaches its destination, the Consignee submits a Report of Receipt to notify the delivery to the involved actors as described in section III.I.4 Submission of Report of Receipt (UC2.06). If the Report of Receipt declares shortages or excesses, then the Consignor and/or the Consignee may provide explanations by submitting a complementary explanations (IE871: C_SHR_EXP) message. The submission of explanations is optional for both the Consignor and the Consignee and may occur several times by the same actors.

The scenarios of this use case require the Consignee having submitted a Report of Receipt describing shortages or excesses. In addition, the Consignor may submit complementary explanations after confirmation of exit of exported goods. At least one of the following scenarios must have been concluded:

Section	Scenario	State
III.I.4.1	Delivery Accepted	Delivered
III.I.4.2	Delivery Refused	Refused
III.I.4.3	Delivery Partially Refused	Partially Refused

Table 14: Scenarios that must have been previously completed in order to submit complementary explanations

One scenario is described in this use case using actor instantiations. The “*Consignor / Consignee*” instance of the *Economic operator* actor should be considered as either the *Consignor* or the *Consignee* actor. The “*dispatch / destination*” and the “*destination / dispatch*” instance of the *MSA application* actor should be considered as either the *MSA dispatch application* or the *MSA destination application* actor.

Note that this use case can be performed in two separate scenarios depending on whether the Consignor or Consignee has initiated the use case by submitting complementary explanations. The message exchange protocol for both scenarios is identical.

If the scenario is initiated by the *Consignor*, then “*dispatch / destination : MSA application*” should be considered as the *MSA dispatch application* and the “*destination / dispatch : MSA application*” should be considered as the *MSA destination application*. If the scenario is initiated by the *Consignee*, then “*dispatch / destination : MSA application*” should be considered as the *MSA destination application* and the “*destination / dispatch : MSA application*” should be considered as the *MSA dispatch application*.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The *Consignor / Consignee* Economic operator submits complementary explanations (IE871: C_SHR_EXP) to the *dispatch / destination MSA application* for an e-AD that shortages or excesses have been declared.

The *dispatch / destination MSA application* receives the complementary explanations (IE871: C_SHR_EXP) and performs validation. Upon successful validation (exceptional cases are described in Sub-Section III.VI), the *dispatch / destination MSA application* forwards the incoming complementary explanations (IE871: C_SHR_EXP) message to the *destination / dispatch MSA application*.

The *destination / dispatch MSA application* validates the message structure of the received complementary explanations (IE871: C_SHR_EXP).

The scenario for sending complementary explanations is depicted in Figure 48 and Figure 49:

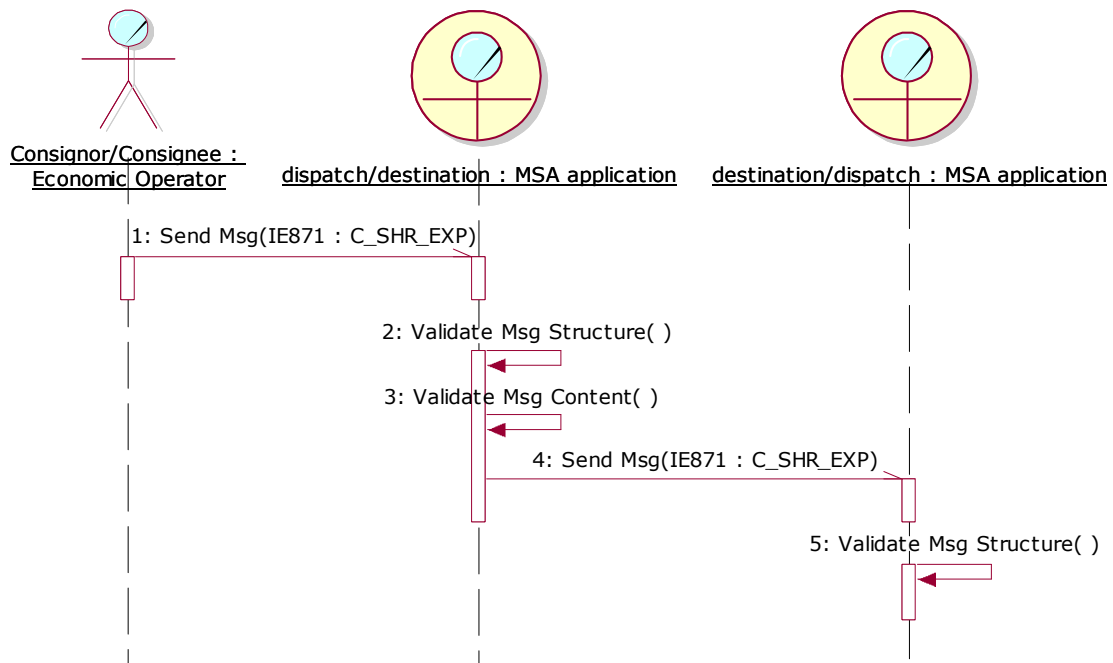


Figure 48: TSD - Complementary explanations submission

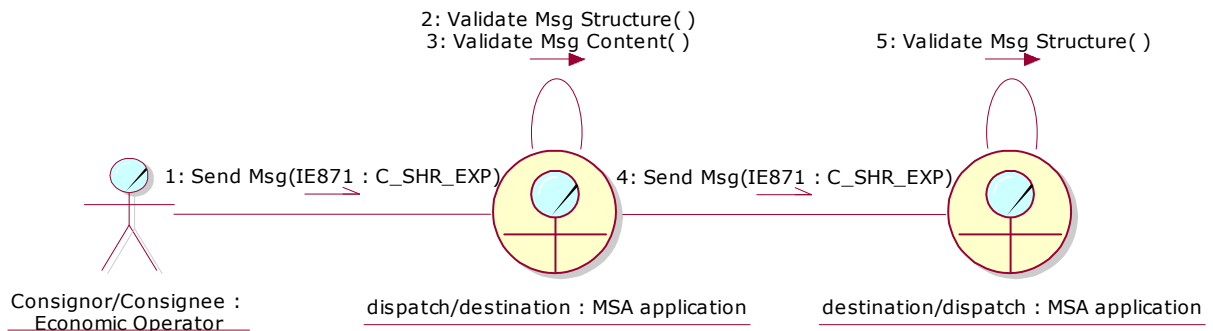


Figure 49: CLD - Complementary explanations submission

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.I.11 Information on intended claims (UC2.13)

This use case describes the notification of the MSA of Dispatch by any MSA application claiming the recovery of duties for a movement that shortages have been ascertained. The process within the scope of EMCS includes only the exchange of a basis for recovery of duties (IE861: C_RES_DAT) message; the actual procedures for the recovery of duties are outside the scope of EMCS. The use case ensures that the MSA of Dispatch will have all claims readily available including the cases where concurrent claims have been issued for a particular movement.

The MSA of Destination that estimates lost excise duties (shortages) may send an IE861 to the MSA of Dispatch for registering its claims. In case the MSA that claims duties is neither the MSA of Dispatch nor the MSA of Destination, the actor third MSA application will be utilised. The recovery of duties (IE861: C_RES_DAT) message is created when an Excise officer is using the application to submit the message on behalf of the MSA claiming duties. The details of this interaction are not included in the described scenarios as they are a national matter.

The scenarios of this section describe the “Information on intended claims (UC2.13)” use case, through the following scenarios:

- duties claimed by the MSA of Destination (see III.I.11.1);
- duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination (see III.I.11.2).

III.I.11.1 Duties claimed by the MSA of Destination

The MSA destination application submits a basis for recovery of duties (IE861: C_RES_DAT) message to the MSA dispatch application. It shall be noted that in general, when the claiming MSA is the MSA of Destination, the download of the e-AD (see section III.I.12 Download of an e-AD (UC2.51)) in order to obtain additional information, with respect to the movement, before sending the analysis (IE861: C_RES_DAT) message is optional. However, if deemed necessary by the Excise officer performing the investigation, that the results of the analysis (IE861: C_RES_DAT) contains Event Report or Control Report references that are not available in the MSA of Destination application, then the e-AD download must be performed. This ensures that the referenced Event Reports or Control Reports exist and it will enable the MSA of Destination application to validate the existence of the referenced Event Reports or Control Reports. Furthermore, optionally the “General query to retrieve an e-AD (UC2.52)” (see section III.I.13 General query to retrieve an e-AD (UC2.52)) can also be used to retrieve information.

Upon the reception, the MSA dispatch application validates the information message (IE861: C_RES_DAT). Upon successful validation (exceptional cases are described in Sub-Section III.VI), the MSA dispatch application registers the incoming information message.

The scenario for duties claimed by the MSA of Destination is depicted in Figure 50 and Figure 51:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

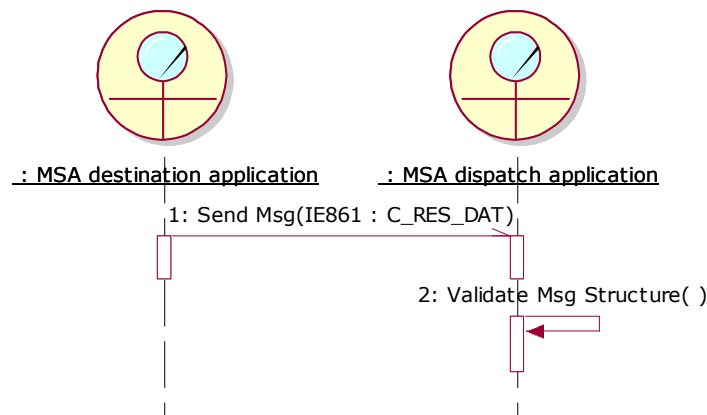


Figure 50: TSD - Duties claimed by the MSA of Destination

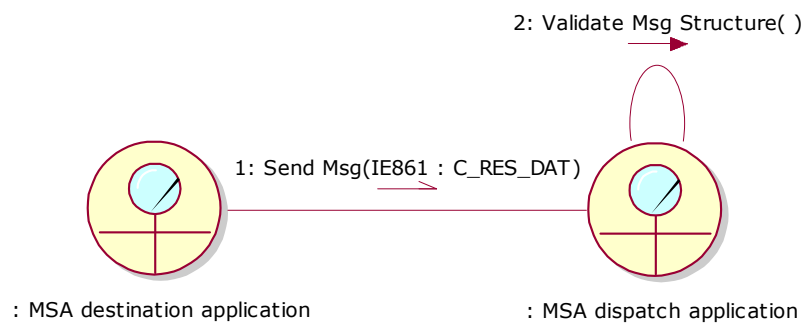


Figure 51: CLD - Duties claimed by the MSA of Destination

III.I.11.2 Duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

A third MSA application submits a basis for recovery of duties (IE861: C_RES_DAT) message to the MSA dispatch application. In case a third MSA requires additional information, with respect to the movement, it obtains this information through use case “Download of an e-AD (UC2.51)” (see sections III.I.12 Download of an e-AD (UC2.51)). More specifically, it is mandatory for a third MSA that has not been involved in the movement (i.e. it is not a former MSA of Destination) to always download the e-AD before sending the analysis (IE861: C_RES_DAT) message. If the third MSA is a former MSA of Destination and the Report of Receipt ascertaining shortages has been sent by the same MSA, then the download of the e-AD is optional, but recommended. However, if the third MSA is a former MSA of Destination and the Report of Receipt ascertaining shortages has not been sent by the same MSA, then the download of the e-AD is mandatory. Also, if deemed necessary by the Excise officer performing the investigation, that the results of the analysis (IE861: C_RES_DAT) contains Event Report or Control Report references that are not available in the third MSA application, then the e-AD download must be performed. This ensures that the referenced Event Reports or Control Reports exist and it will enable the third MSA application to validate the existence of the referenced Event Reports or Control Reports. Furthermore, optionally the “General query to retrieve an e-AD (UC2.52)” (see section III.I.13 General query to retrieve an e-AD (UC2.52)) can also be used to retrieve information.

Upon the reception, the MSA dispatch application validates the information message (IE861: C_RES_DAT). Upon successful validation (exceptional cases are described in Sub-Section III.VI), the MSA dispatch application registers the incoming information message.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The scenario for duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 52 and Figure 53:

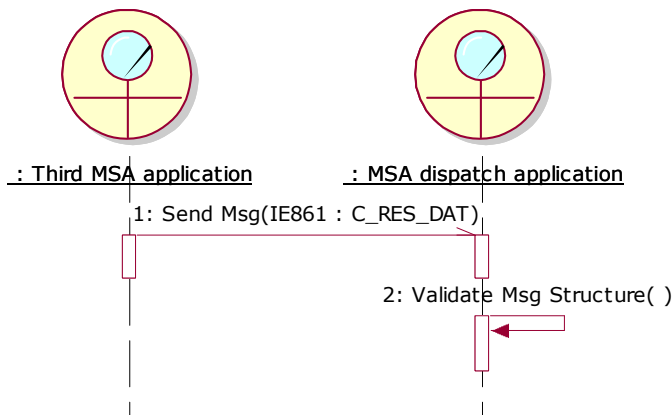


Figure 52: TSD - Duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

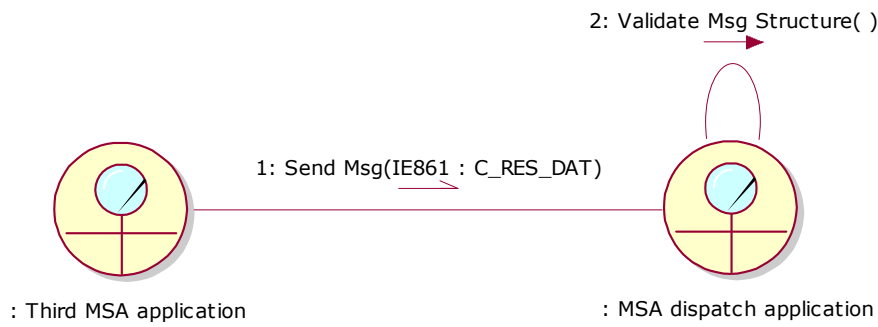


Figure 53: CLD - Duties claimed by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

III.I.12 Download of an e-AD (UC2.51)

This use case describes the message exchange protocol for requesting to download an e-AD. Two different scenarios of this use case are described. The first scenario describes the case where the e-AD is available on-line in which case the MSA of Dispatch responds with a data packaging message (IE934: C_PAC_DAT) that includes all business messages comprising the movement history. The second scenario describes the case where the e-AD is not available on-line. In this case the Requesting MSA may follow the scenarios described in IV.III.1 Request for history information in order to request the historical e-AD data.

The same correlation principle as explained in detail in Section III.VI.2 is applied to the IE934 message of the Download of an e-AD (UC2.51) scenario, for consistency and enhanced correlation. In particular, the “Correlation Identifier” Data Item of the IE905 and IE934 should be equal to the “Message Identifier” of the IE904.

III.I.12.1 Download of an e-AD successful

This scenario *shall* be supported by a MSA in order to allow its MSA Officials to download from the MSA of Dispatch all records concerning a given e-AD. In this scenario, it is assumed that the MSA of Dispatch application is different than the Requesting MSA application and that all validations of the incoming messages pass successfully.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The e-AD consultation is explicitly initiated by a MSA Official, who provides the ARC of the queried e-AD accompanied by the last known sequence number for the specific ARC indicating also that he/she is interested in the movement history.

The Requesting MSA application builds and sends the query of the MSA Official to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) via the Status Request (IE904: C_STD_REQ) message, including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA of Dispatch. It is noted that in case that the Requesting MSA has not received any IE that is related to the concerned ARC, then it has to set the status to “None” and the last message received from the MSA of Dispatch to “None”. Furthermore, the Requesting MSA application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history. If the latter is not indicated then only the status of the movement will be returned back from the initiating MSA.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI).

The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the Requesting MSA application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA of Destination for the specific ARC (e.g. the IE818). Following the submission of the IE905, the MSA dispatch application also submits to the Requesting MSA application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history, that is the business messages that have been exchanged in the Common Domain and have been successfully processed by the MSA dispatch application.³

More specifically the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC of the specific movement in the MSA dispatch application. It is the same as the ARC of the request (IE904) message;
- The “Sequence Number”: It includes the latest Sequence Number for the specific ARC in the Requesting MSA application;
- The “Status”: It includes the State for the specific ARC in the Requesting MSA application;
- The “Last Received Message Type”: It includes the last message received from the MSA of Destination for the specific ARC.

³ Any technical Status Request/Response messages will not be incorporated in the movement history except for the IE905 used for manual closure of a movement (see III.VI.3 Manual Closing of the Movement).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The IE934 message should have the following values:

- The “Common Message Header” should conform to the guidelines of Section VII.I.4 and consist of the following values in the relevant Data Items:
- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items should conform to the guidelines of Section VII.I.4;
- The “Message identifier”: This is a “Required” Data Item generated by the sending application to uniquely identify the information exchange;
- The “Correlation identifier”: In the IE934 message the “Correlation identifier” Data Item should have the same value as the “Correlation identifier” of the IE905 and the “Message identifier” of the IE904 message for correlating the response (IE905 and IE934) messages with the request (IE904) message.
- The “Body” contains the actual business message with the following Data Items:
 - The “Technical Message Type”: It includes the number of the specific messages that are part of the movement history, identified by the ARC of the request (IE904) message;
 - The “Message Data”: It includes the message data of the messages comprising the movement history for the specific ARC in the MSA dispatch application.

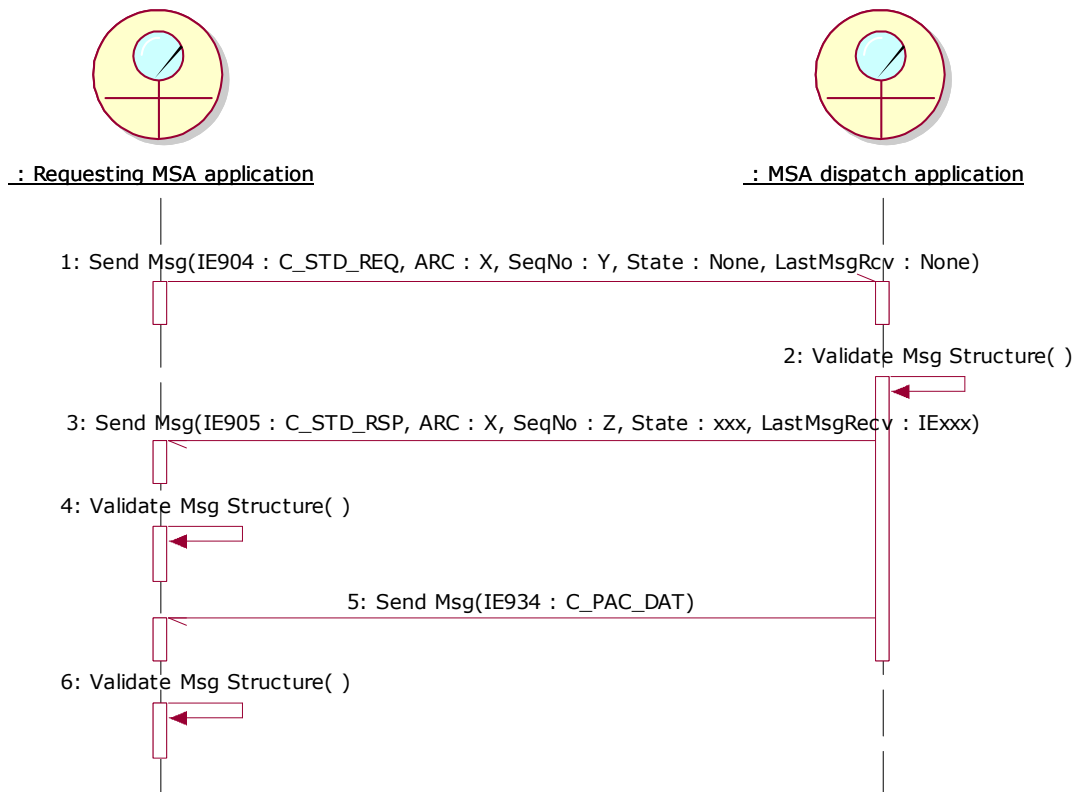


Figure 54: TSD - e-AD information downloaded successfully

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

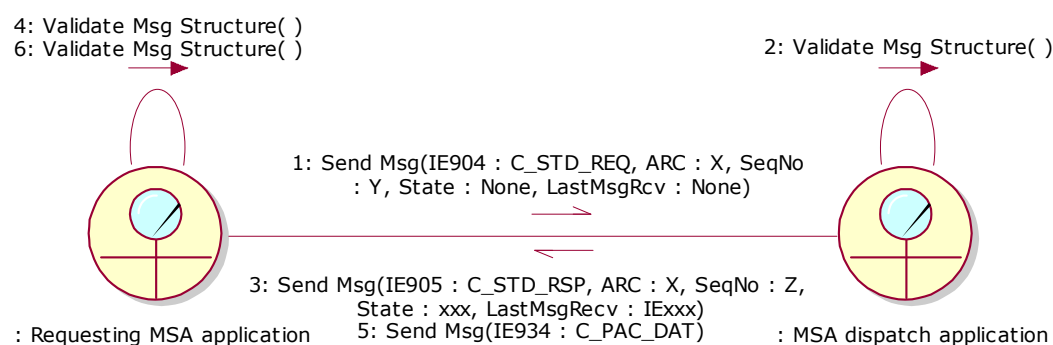


Figure 55: CLD - e-AD information downloaded successfully

III.I.12.2 Download of an e-AD failed

In this scenario, contrary to the previous section, no e-AD is found. A Status Response (IE905: C_STD_RSP) with “Status = None” and last message set to “None” is sent back to the Requesting MSA application. The ARC in the IE905 is the same as the one included in the received IE904 and the sequence number is set to the value “1” (since the ARC is not known at MSA of Dispatch). This means either that the ARC is invalid or that the commonly agreed time for consultation of movement data has passed, and therefore the e-AD has been archived and is no more available on-line. When deemed necessary a MSA may request archived/historical data to be retrieved by following the scenarios described in IV.III.1 Request for history information.

More specifically the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC specified in the request (IE904) message;
- The “Sequence Number”: It includes the value “1”, since the specific ARC is unknown to the MSA dispatch application;
- The “Status”: It includes the value “None”, since the specific ARC is unknown to the MSA dispatch application;
- The “Last Received Message Type”: It includes the value “None”, since the specific ARC is unknown to the MSA dispatch application.

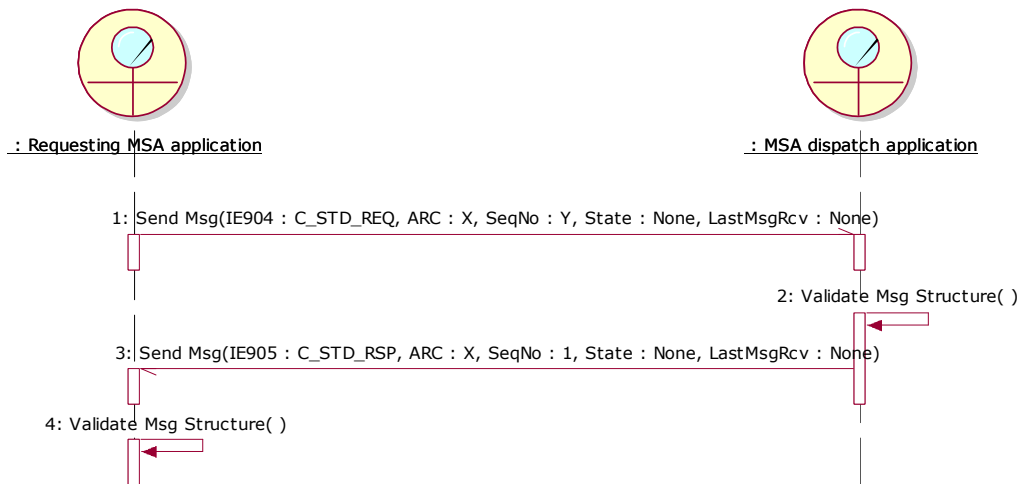


Figure 56: TSD - Download of an e-AD failed

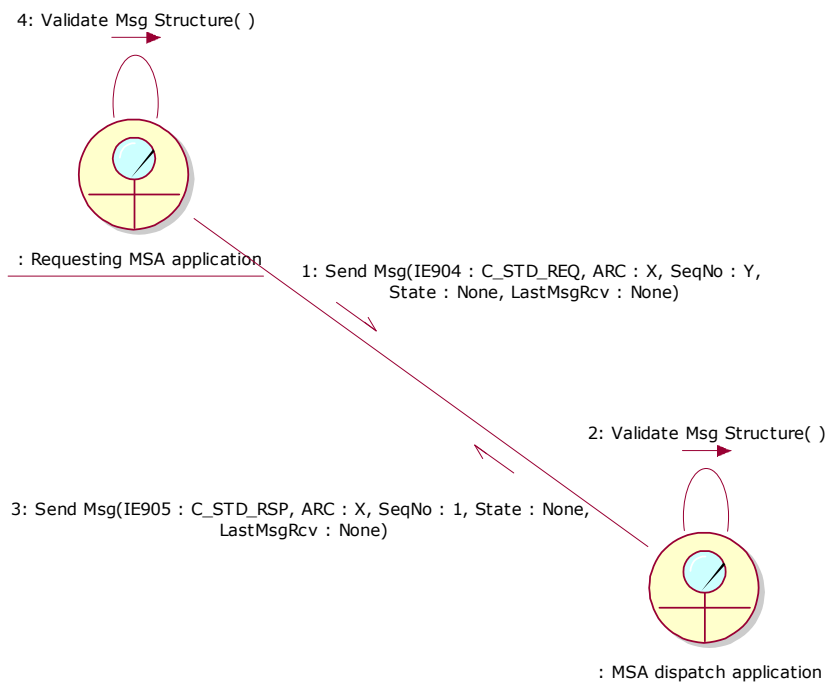


Figure 57: CLD - Download of an e-AD failed

III.I.13 General query to retrieve an e-AD (UC2.52)

This scenario describes the message exchange protocol between a MSA application and an initiator MSA application (which has initiated the e-AD) when a MSA Official wants to retrieve an e-AD, which has been initiated by a MSA application located in a different MSA from that of request.

The national MSA application (e-AD requestor) sends a query (IE701: C_REQ_SUB) to the supposed MSA initiator of the requested e-AD. Upon the reception of the e-AD query (IE701: C_REQ_SUB), the Initiator MSA application validates successfully the received message and responds with a list of e-ADs (IE821: C_LST_VAL) if one or more e-ADs are retrieved. The Initiator MSA application responds with a refusal message (IE702: C_REQ_REF) for the cases where:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- No e-AD matches the search criteria;
- The maximum limit of retrieved e-ADs has been reached;
- The only one e-AD that matches the query has been archived.

It should be noted that any search performed by the Initiator MSA application against the ARC value should be case insensitive and therefore any ARCs that differ only in their capitalization should be considered as identical.

The exceptional cases of the unsuccessful validation of the e-AD query message received by the initiator MSA application are described in Sub-Section III.VI.

If the e-AD is archived and the MSA Official still wants to retrieve this e-AD, then Section IV.III.1 “Request for history information” that includes use case UC3.29 should be considered.

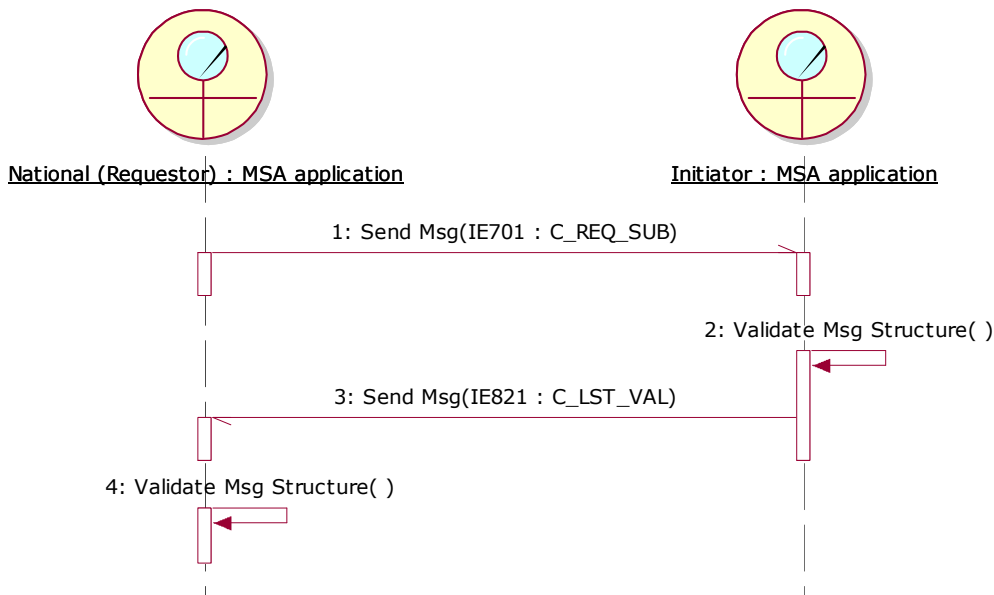


Figure 58: TSD - Successful retrieval of e-AD(s)

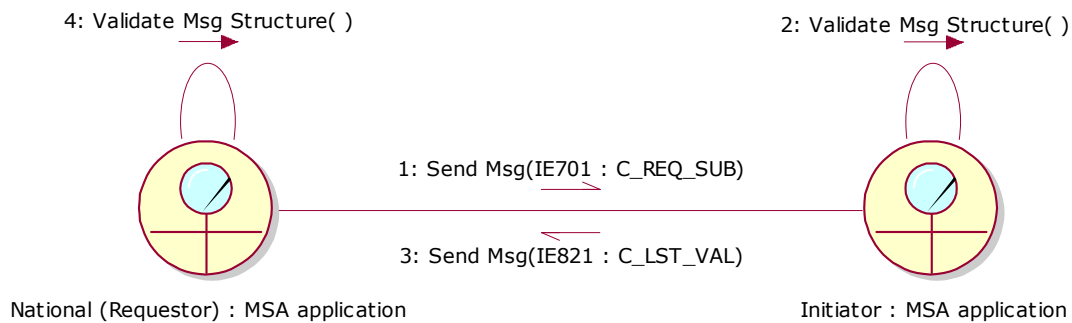


Figure 59: CLD - Successful retrieval of e-AD(s)

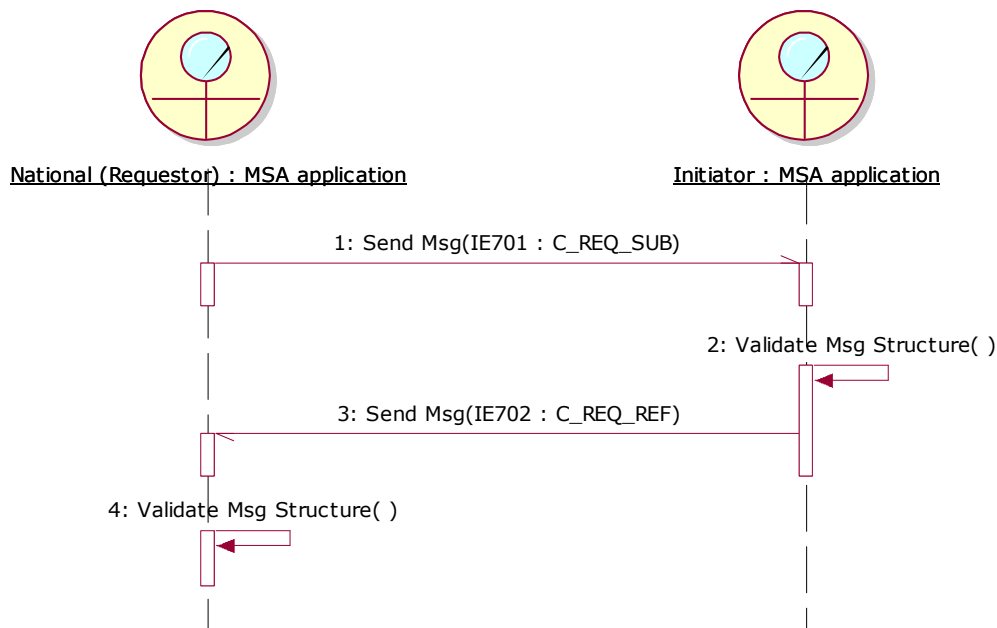


Figure 60: TSD - No movement found or limit exceeded

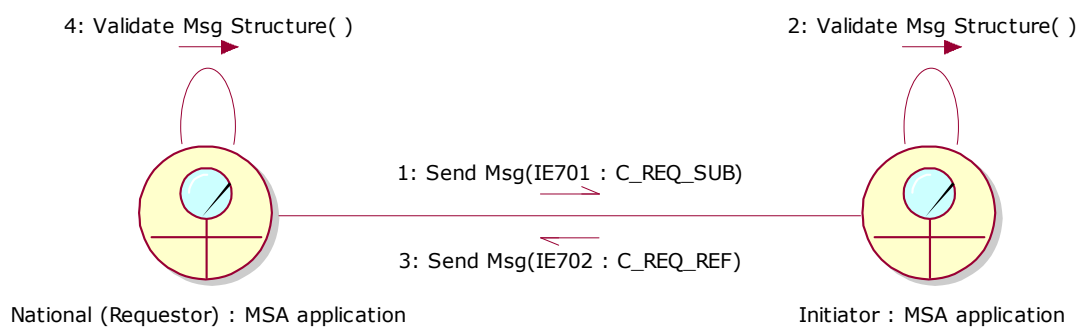


Figure 61: CLD - No movement found or limit exceeded

Sub-Section III.II Export Scenarios

This section aims to specify all possible message exchange protocols involved in the Export cases. The identification of these scenarios has been based on the “exportation of goods” use cases in FESS [A1], which define the exportation of excise goods moving under duty suspension arrangements outside the European Community by triggering:

- local clearance at export, where the Consignor submits both the e-AD and the export declaration in his own premises; or
- export operation at office of export, where the Consignor submits only the e-AD, whereas the export declaration is submitted at the office of export (possibly in a different Member State).⁴

It shall be noted that:

⁴This case is a rare scenario to take place in real life.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- These scenarios assume the availability of ECS Phase 3, hence full automation of the Export cases. This is, however, a recommended implementation. The integration of EMCS and ECS and the implementation details remain a national matter (see also the SD [A2]);
- Any communication outside the scope of the EMCS application (such as, the information exchanged between the Consignor or forwarding agent and the Customs Export Application) is not in the interest of this document.

III.II.1 Local Clearance at Export

In the case of “Local Clearance at Export”, the Consignor submits both the e-AD and the export declaration at his own premises. Hence, the Member State of export is always the Member State of dispatch. It shall be noted that:

- The actor *MSA dispatch application* in the Sequence Diagrams is the *MSA dispatch/export application* appearing in FESS [A1]. In this case there is no Consignee and the destination fields in the e-AD are not applicable.
- For simplification reasons the Sequence Diagrams depict the submission and registration of only one draft e-AD. However, it is possible that an export declaration includes more than one ARC. In this case, all “e-AD” references should be read as “all concerned e-ADs”.

All the “Local Clearance at Export” scenarios refer to an e-AD with a sequence number “1”. Therefore, all the IE messages exchanged in the “Local Clearance at Export” bearing the sequence number, namely the IE801, IE818, IE829 and IE839 messages include a sequence number “1”. It is noted that the IE839 bears the ARC and sequence number only when the reason for the negative cross-checking (<Rejection Reason Code>) is ‘2: The contents of the e-AD does not match with export data’.

III.II.1.1 Local Clearance at Export followed by Export Confirmation of Exit (UC2.44)

In this scenario, all message validations pass successfully. The Anticipated Export Record (IE501: C_AER_SND) is cross-checked successfully against the e-AD and the exit from the Community is confirmed.

III.II.1.1.1 Local Clearance at Export (UC2.44)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “*Accepted for export*”.

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the Consignor submits the export declaration to the Customs Export Application (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

When the export is released by the Customs Export Application, the Anticipated Export Record (IE501: C_AER_SND) is forwarded to the MSA dispatch application.

Upon the reception of the Anticipated Export Record (IE501: C_AER_SND) from the Customs Export Application, the MSA dispatch application cross-checks it against the e-AD. The cross-checking is positive and the MSA dispatch application builds a notification message (IE829: C_EXP_NOT) and sends it back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “*Exporting*” and the timer TIM_EAD is initiated to expire at the expected date of exit (date of dispatch + journey time).

The MSA dispatch application is waiting for the discharge to take place when the exit from the Community is completed.

III.II.1.1.2 Export Confirmation of Exit (UC2.46)

The MSA dispatch application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is accepted (IE518/Control Result Code: A1, A2, A4), the MSA dispatch application builds the Report of Export (IE818) reporting exit acceptance (IE818/Global conclusion of receipt: 21, 22) and forwards it to the Consignor. Moreover, the state of the movement at the MSA of Dispatch is updated from “*Exporting*” to “*Delivered*”. Finally, if the TIM_EAD timer has not expired, the MSA dispatch application stops it, otherwise it resets the flag raised locally at its expiration.

The scenario with the acceptance of delivery is depicted in Figure 62 and Figure 63:

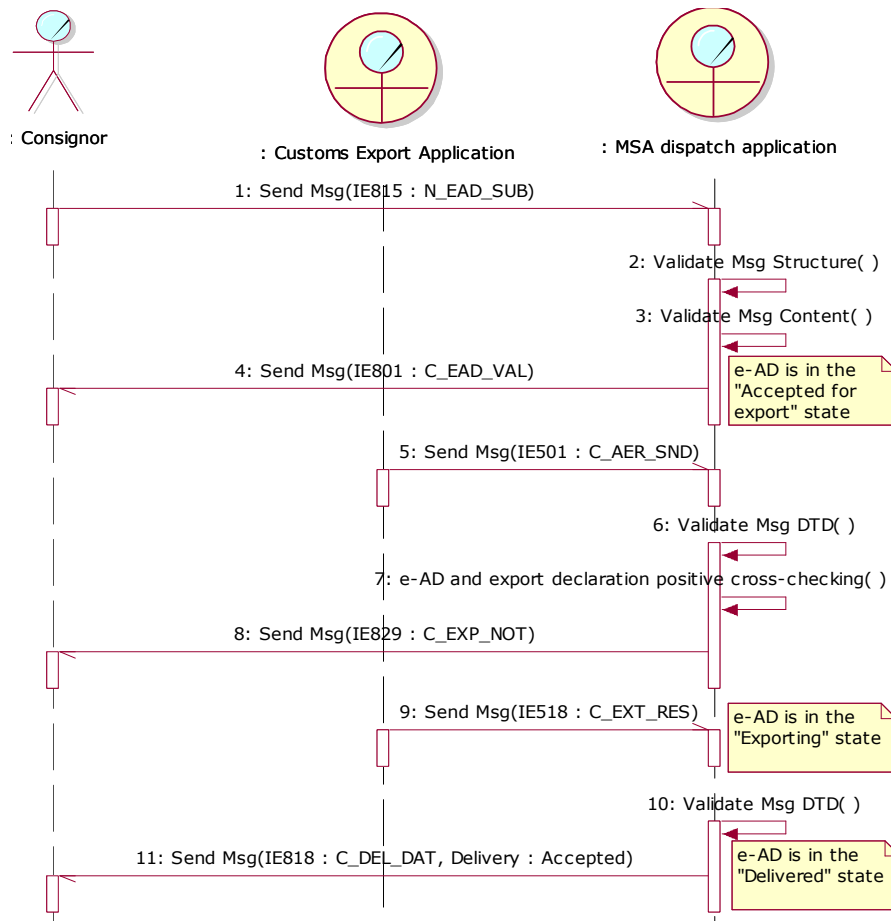


Figure 62: TSD - Local Clearance at Export followed by Export Confirmation of Exit

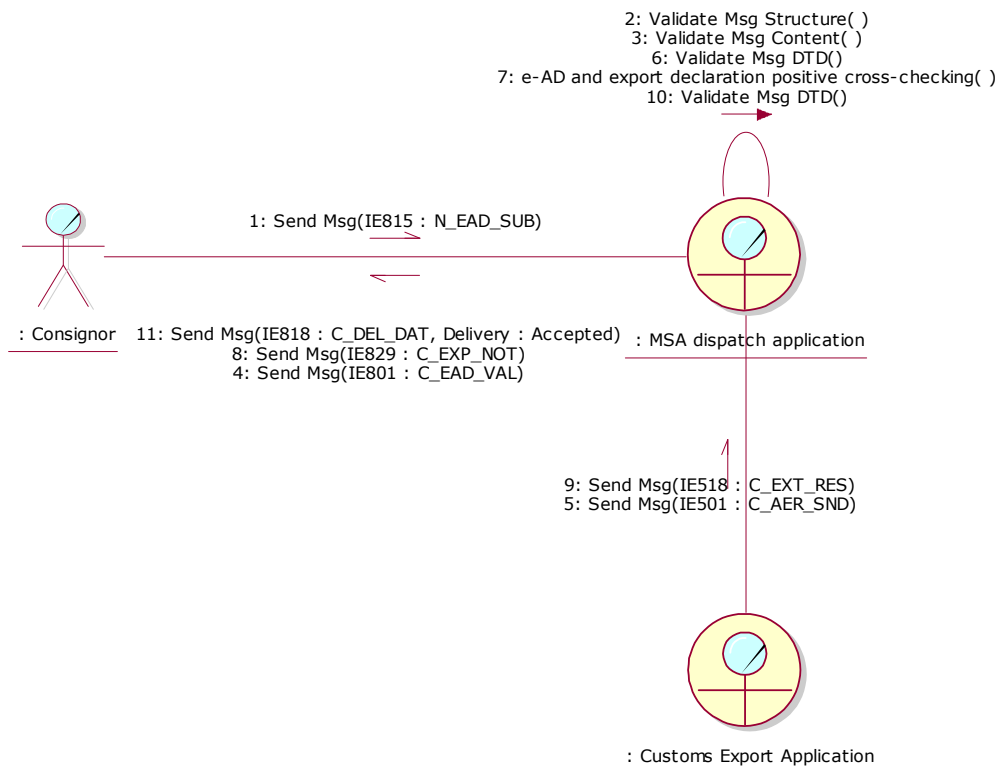


Figure 63: CLD - Local Clearance at Export followed by Export Confirmation of Exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.1.2 Local Clearance at Export followed by Export Cancellation of Exit

In this scenario, all message validations pass successfully. The Anticipated Export Record (IE501: C_AER_SND) is cross-checked successfully against the e-AD but the exit is refused.

III.II.1.2.1 Local Clearance at Export (UC2.44)

The “Local Clearance at Export” is exactly the same as in Section III.II.1.1.1 Local Clearance at Export (UC2.44). Hence, the e-AD is in the “*Exporting*” state, the TIM_EAD has been initiated and the MSA dispatch application is waiting for the discharge to take place when the exit from the Community is completed.

III.II.1.2.2 Export Cancellation of Exit (UC2.46)

The MSA dispatch application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is refused (IE518/Control Result Code: B1), the MSA dispatch application builds the Report of Export (IE818: C_DEL_DAT) reporting exit refusal (IE818/Global conclusion of receipt: 23) and forwards it to the Consignor. Moreover, the state of the movement at the MSA of Dispatch is updated from “*Exporting*” to “*Refused*”. Finally, it starts the TIM_CHS timer waiting for the change of destination from the Consignor.

III.II.1.2.3 Change of Destination (UC2.05)

Upon receipt of the exit refusal (IE818/Global conclusion of receipt: 23), the Consignor issues a change of destination, as described in the scenario of Section III.I.5 Change of Destination (UC2.05).

The scenario with the refusal of delivery is depicted in Figure 64 and Figure 65:

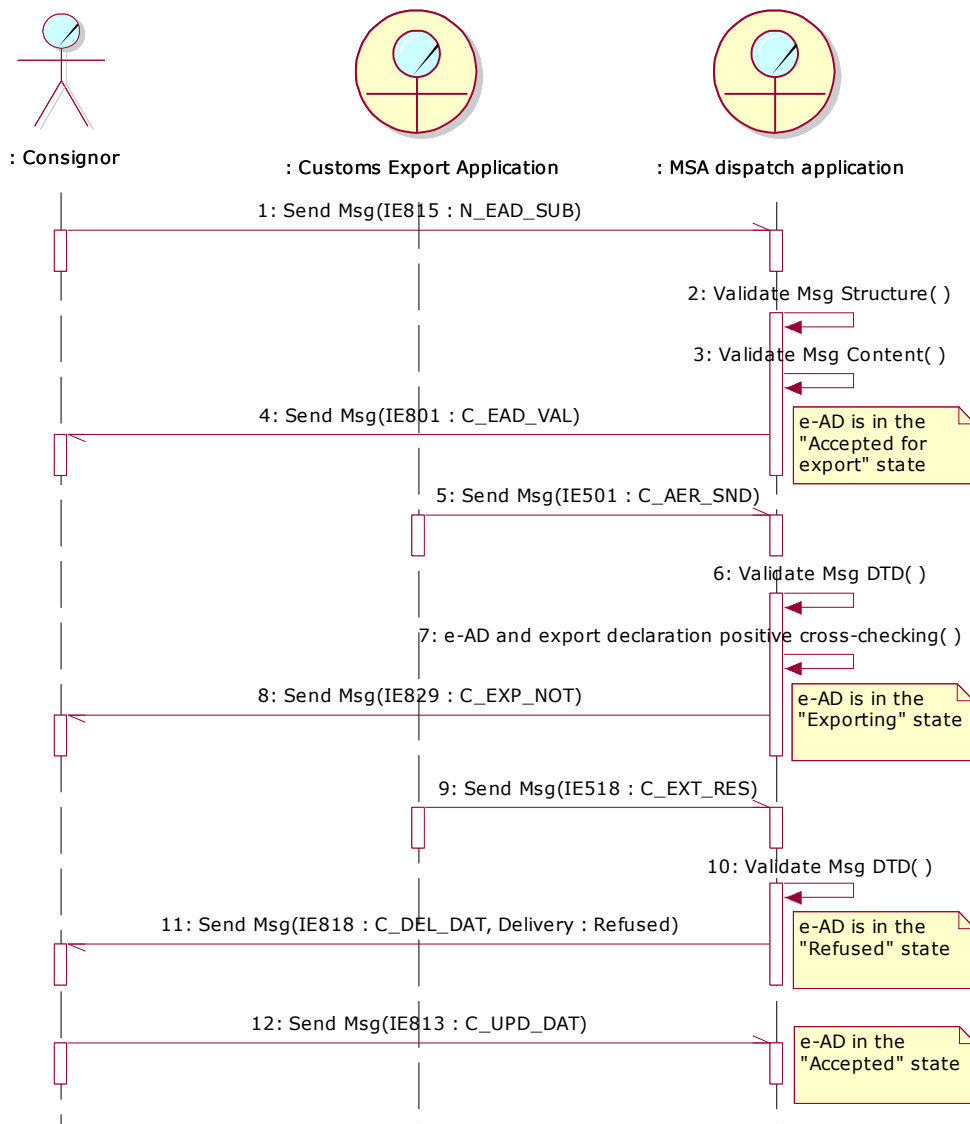


Figure 64: TSD - Local Clearance at Export followed by Export Cancellation of Exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

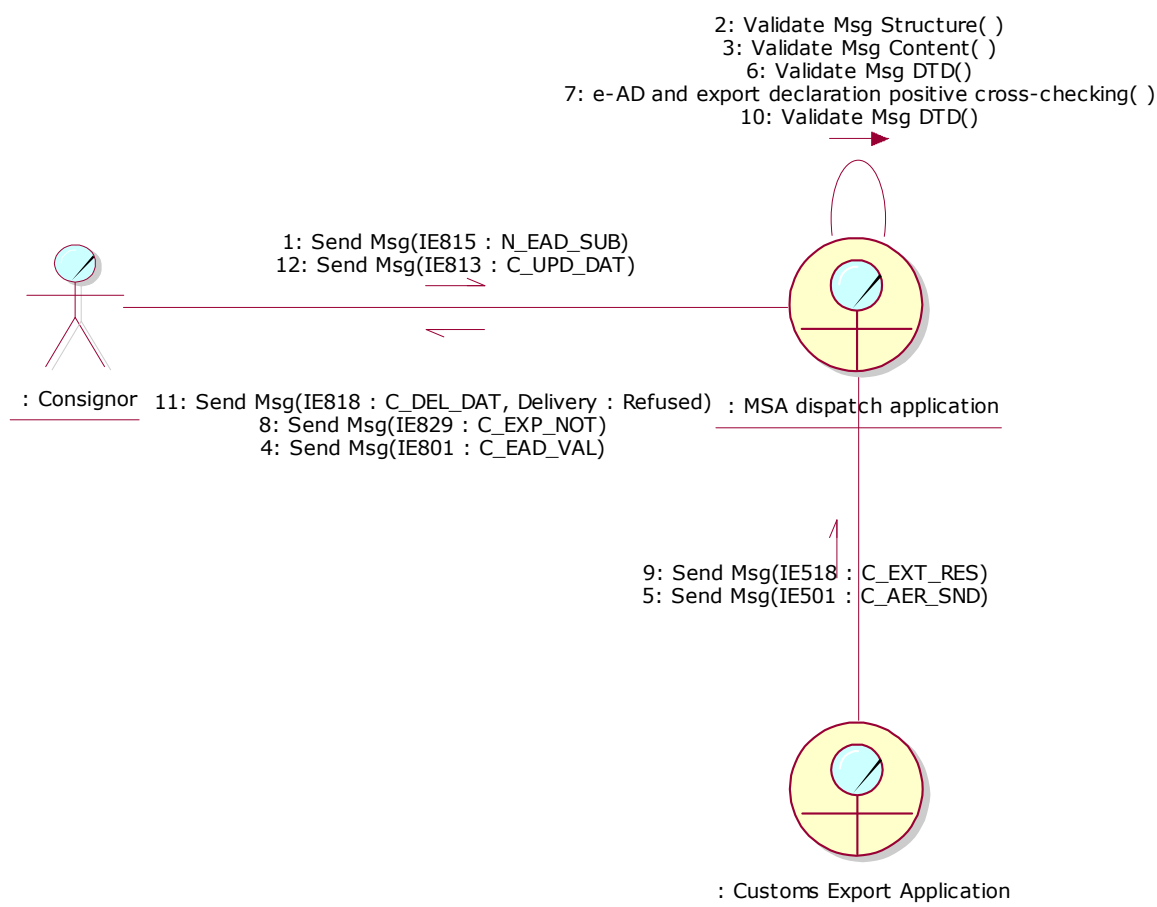


Figure 65: CLD - Local Clearance at Export followed by Export Cancellation of Exit

III.II.1.3 Local Clearance at Export followed by export declaration cancellation and Change of Destination

In this scenario, all message validations pass successfully. The Anticipated Export Record (IE501: C_AER_SND) is cross-checked successfully against the e-AD but the Consignor cancels the export declaration and submits a Change of Destination.

III.II.1.3.1 Local Clearance at Export (UC2.44)

The “Local Clearance at Export” is exactly the same as in Section III.II.1.1.1 Local Clearance at Export (UC2.44). Hence, the e-AD is in the “Exporting” state, the TIM_EAD has been initiated and the MSA dispatch application is waiting for the discharge to take place when the exit from the Community is completed.

III.II.1.3.2 Change of Destination (UC2.05)

The Consignor cancels the export operation in ECS while the e-AD is in the “Exporting” state. Following the export operation cancelation in ECS, the Consignor submits a change of destination, as described in the scenario of Section III.I.5 Change of Destination (UC2.05).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The change of destination should be performed since the cancellation of the export operation in ECS is not foreseen to be communicated to EMCS. After the issuance of the change of destination by the Consignor, the state of the e-AD in EMCS changes from “*Exporting*” to “*Accepted*” at the MSA of Dispatch. In case a change of destination is not submitted by the Consignor after the cancellation of the export operation in ECS, then the e-AD in EMCS will remain in the “*Exporting*” state at the MSA of Dispatch, the timer TIM_EAD will expire, and the MSA authorities should take action.

The scenario with the export declaration cancellation by the Consignor and the change of destination is depicted in Figure 66 and Figure 67:

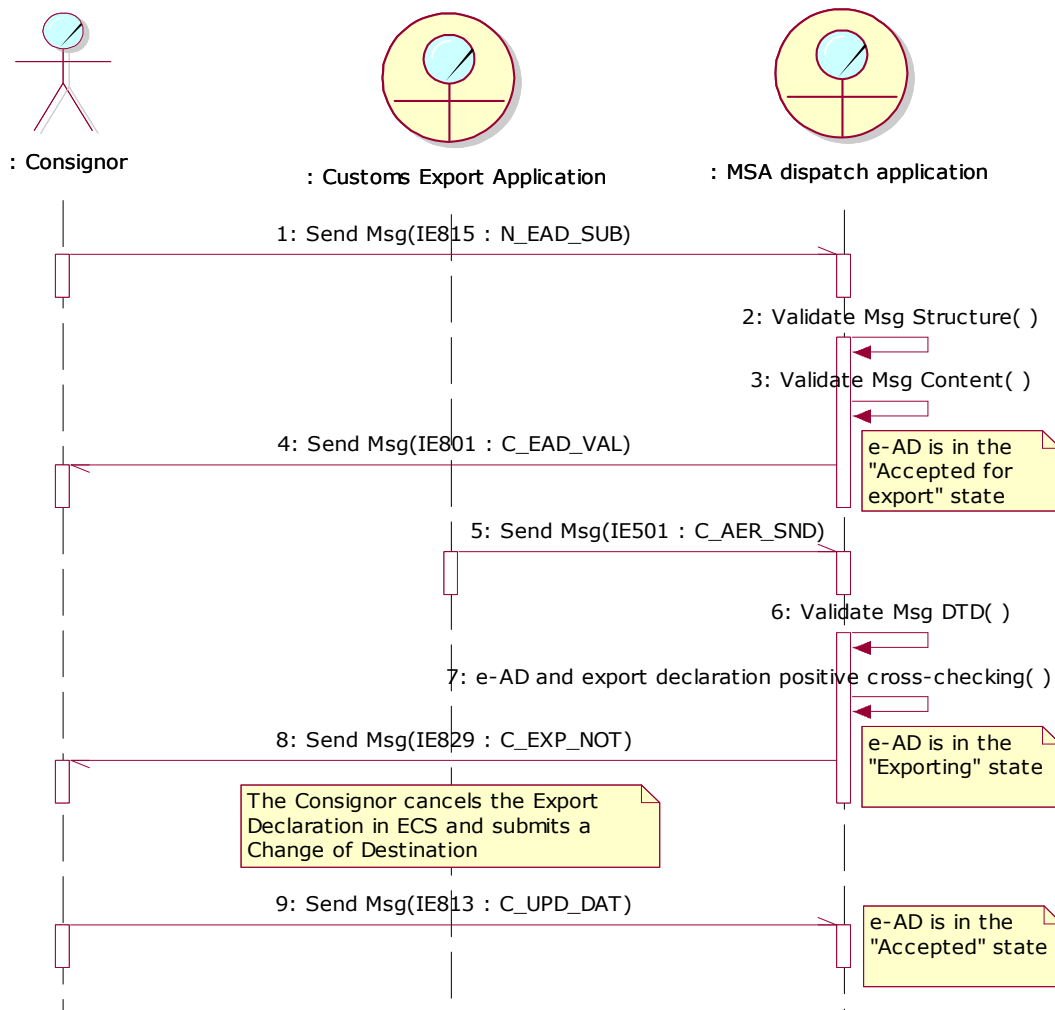


Figure 66: TSD - Local Clearance at Export followed by export declaration cancellation and Change of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

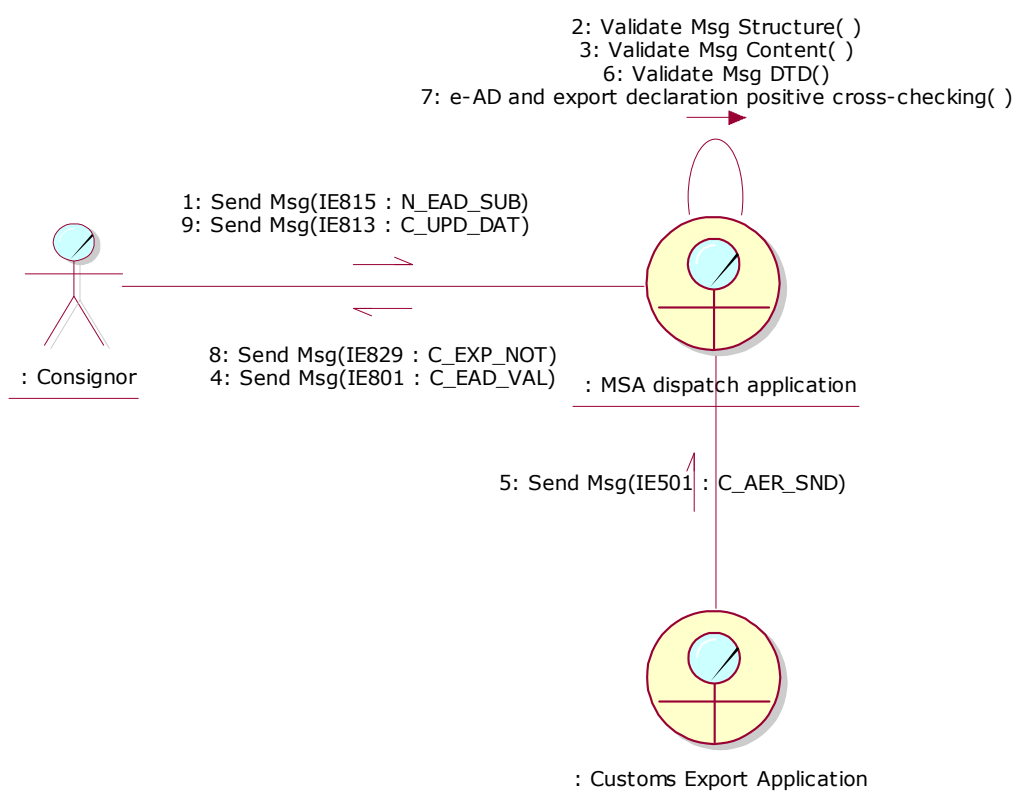


Figure 67: CLD - Local Clearance at Export followed by export declaration cancellation and Change of Destination

III.II.1.4 Local Clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration

In this scenario, the cross-checking between the Anticipated Export Record (IE501: C_AER_SND) and the e-AD is negative. Following, the Consignor cancels the export declaration and submits a new one, which - provided that the movement is released by Customs - will be cross-checked with the existing “Accepted for Export” e-AD.

III.II.1.4.1 Local Clearance at Export (UC2.44)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “Accepted for export”.

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the Consignor submits the export declaration to the Customs Export Application (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams). When the export is released by the Customs Export Application, the Anticipated Export Record (IE501: C_AER_SND) is forwarded to the MSA dispatch application.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon the reception of the Anticipated Export Record (IE501: C_AER_SND) from the Customs Export Application, the MSA dispatch application cross-checks it against the e-AD.

When the cross-checking is negative, the MSA dispatch application builds a rejection message (IE839: C_CUS_REJ) that includes the list of errors found during the cross-checking of the IE501 with e-AD and sends it back to the Consignor to inform him that the submitted e-AD could not be exported. Finally, the state of the movement at the MSA of Dispatch is retained to “*Accepted for export*”.

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the Consignor cancels the export declaration and submits a new one (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

This scenario may be followed by:

- Release by Customs, positive cross-checking and export confirmation of exit (Section III.II.1.1 Local Clearance at Export followed by Export Confirmation of Exit (UC2.44));
- Release by Customs, positive cross-checking and export cancellation of exit (Section III.II.1.2 Local Clearance at Export followed by Export Cancellation of Exit);
- Release by Customs, negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.1.4 Local Clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration);
- Release by Customs, negative cross-checking and e-AD and export declaration cancellation (Section III.II.1.5 Local Clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation);
- No release by Customs (Section III.II.1.6 Local Clearance at Export and movement not released by Customs followed by e-AD cancellation, Section III.II.1.7 Local Clearance at Export and movement not released by Customs followed by submission of new export declaration).

This scenario is depicted in Figure 68 and Figure 69:

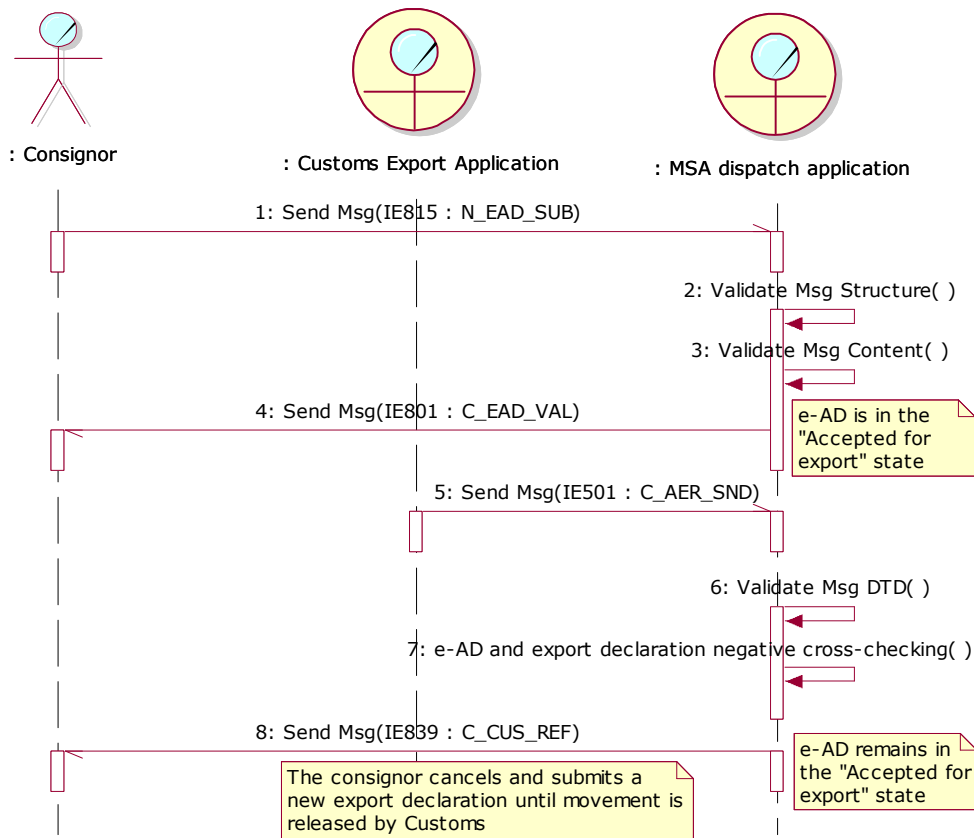


Figure 68: TSD - Local clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration

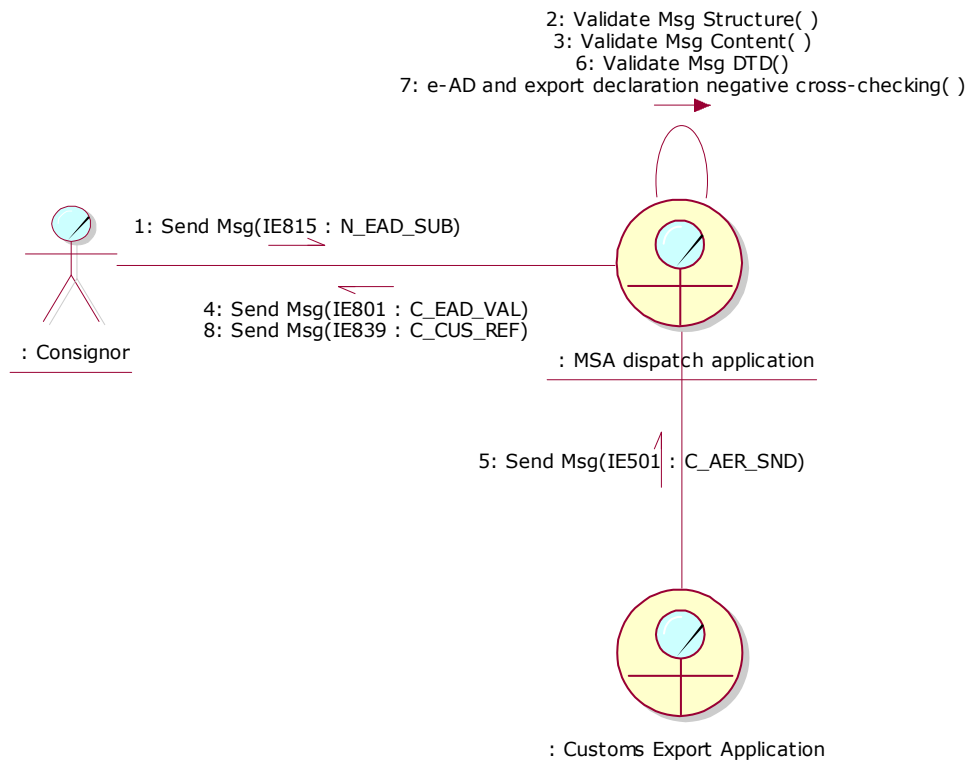


Figure 69: CLD - Local clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.1.5 Local Clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation

In this scenario, the cross-checking between the Anticipated Export Record (IE501: C_AER_SND) and the e-AD is negative. Following, the Consignor cancels the full export operation by cancelling both the export declaration as well as the e-AD.

III.II.1.5.1 Local Clearance at Export (UC2.44)

In this scenario, the cross-checking between the Anticipated Export Record (IE501: C_AER_SND) and the e-AD is negative, as described earlier in the scenario of Section III.II.1.4 Local Clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration. Hence, the movement is in the "Accepted for export" state.

III.II.1.5.2 Cancellation of e-AD (UC2.10)

Upon reception of the rejection message (IE839: C_CUS_REJ) the Consignor cancels both the e-AD and the export declaration. The export declaration is cancelled through the respective message for the Customs application (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

The e-AD is cancelled through the IE810 message and the state of the movement at the MSA of Dispatch is updated from "Accepted for export" to "Cancelled" (see Section III.I.3 Cancellation of an e-AD by the Consignor (UC2.10)).

This scenario is depicted in Figure 70 and Figure 71:

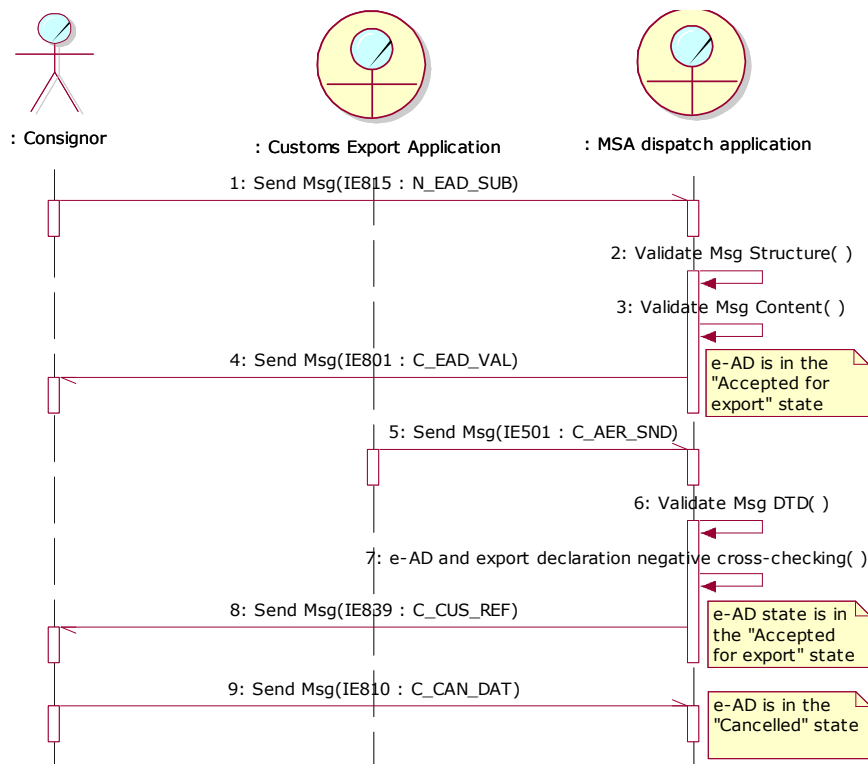


Figure 70: TSD - Local clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

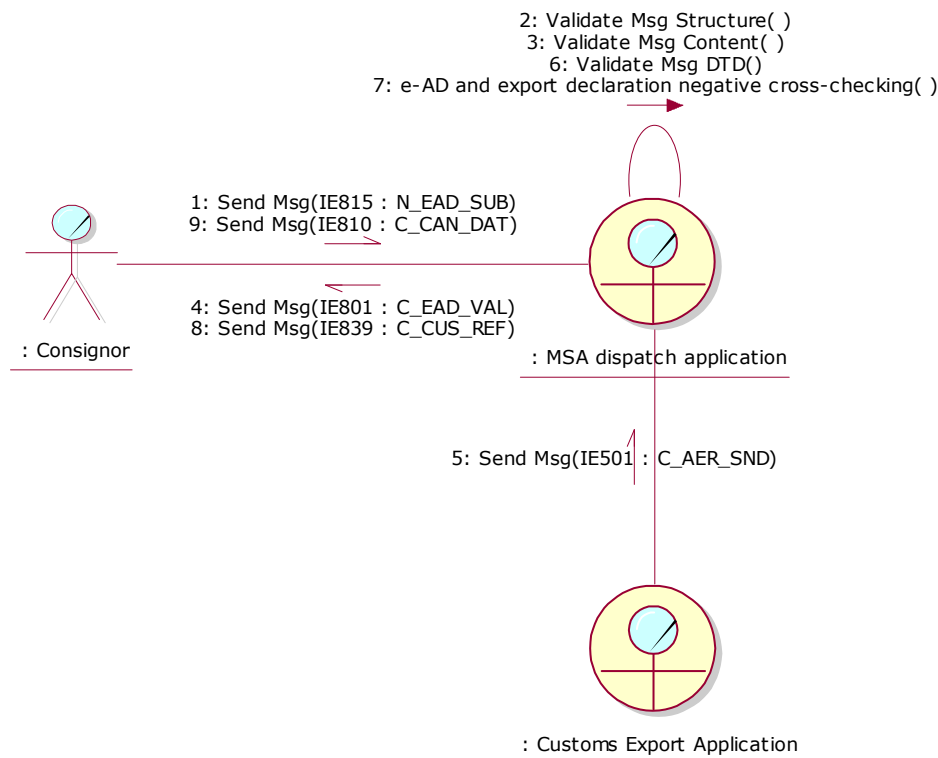


Figure 71: CLD - Local clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation

III.II.1.6 Local Clearance at Export and movement not released by Customs followed by e-AD cancellation

In this scenario, the movement has not been released by Customs. Following, the Consignor cancels the e-AD.

III.II.1.6.1 Local Clearance at Export (UC2.44)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “*Accepted for export*”.

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the Consignor submits the export declaration to the Customs Export Application. However, the Consignor is informed by Customs that the movement could not be released (the communication between the Consignor and the Customs Export Application is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

III.II.1.6.2 Cancellation of e-AD (UC2.10)

Provided that the goods have not left the Place of Dispatch, the Consignor cancels the e-AD through the IE810 message (see Section III.I.3 Cancellation of an e-AD by the Consignor (UC2.10)). The state of the movement is updated from “*Accepted for export*” to “*Cancelled*”

This scenario is depicted in Figure 72 and Figure 73:

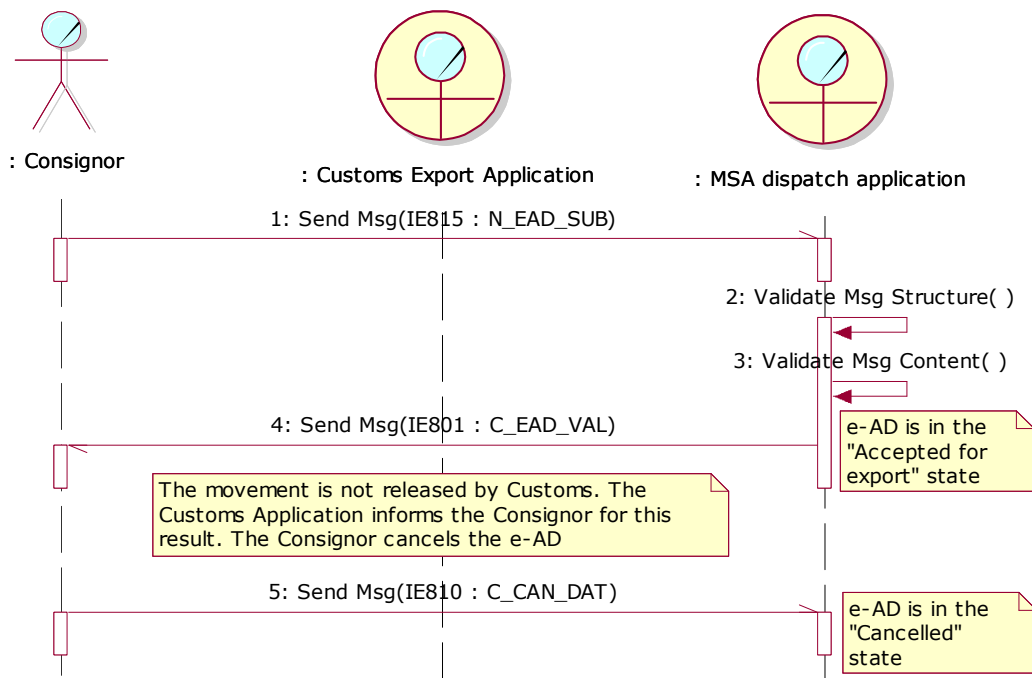


Figure 72: TSD - Local clearance at Export and movement not released by Customs followed by e-AD cancellation

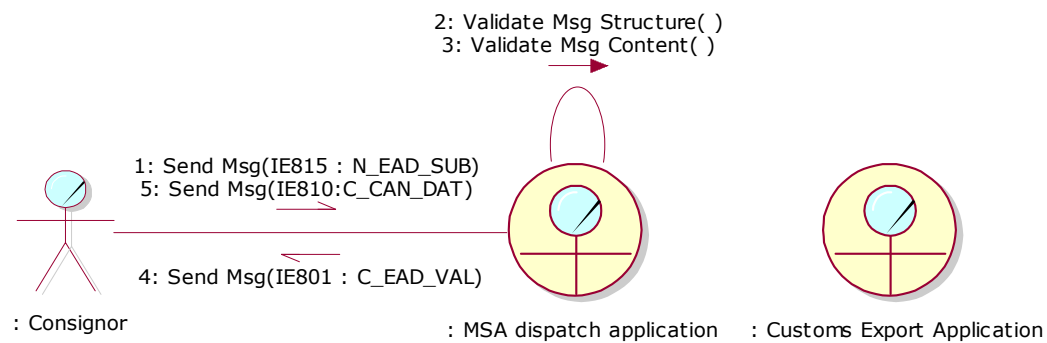


Figure 73: CLD - Local clearance at Export and movement not released by Customs followed by e-AD cancellation

III.II.1.7 Local Clearance at Export and movement not released by Customs followed by submission of new export declaration

The movement has not been released by Customs, as described earlier in the scenario of Section III.II.1.6.1 Local Clearance at Export (UC2.44), and therefore the movement is in the “Accepted for Export” state. Following, the Consignor submits a new export declaration, which - provided that the movement is released by Customs - will be cross-checked with the existing “Accepted for Export” e-AD.

This scenario may be followed by:

- Release by Customs, positive cross-checking and export confirmation of exit (Section III.II.1.1 Local Clearance at Export followed by Export Confirmation of Exit (UC2.44));

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- Release by Customs, positive cross-checking and export cancellation of exit (Section III.II.1.2 Local Clearance at Export followed by Export Cancellation of Exit);
- Release by Customs, negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.1.4 Local Clearance at Export followed by negative cross-checking, export declaration cancellation and submission of new export declaration);
- Release by Customs, negative cross-checking and e-AD and export declaration cancellation (Section III.II.1.5 Local Clearance at Export followed by negative cross-checking and e-AD and export declaration cancellation);
- No release by Customs (Section III.II.1.6 Local Clearance at Export and movement not released by Customs followed by e-AD cancellation, Section III.II.1.6 Local Clearance at Export and movement not released by Customs followed by e-AD cancellation).

This scenario is depicted in Figure 74 and Figure 75:

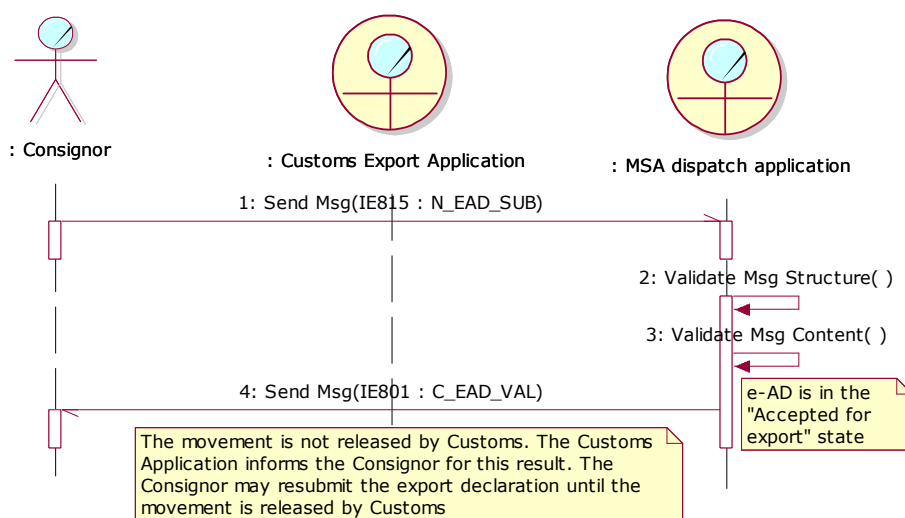


Figure 74: TSD - Local Clearance at Export and movement not released by Customs followed by submission of new export declaration

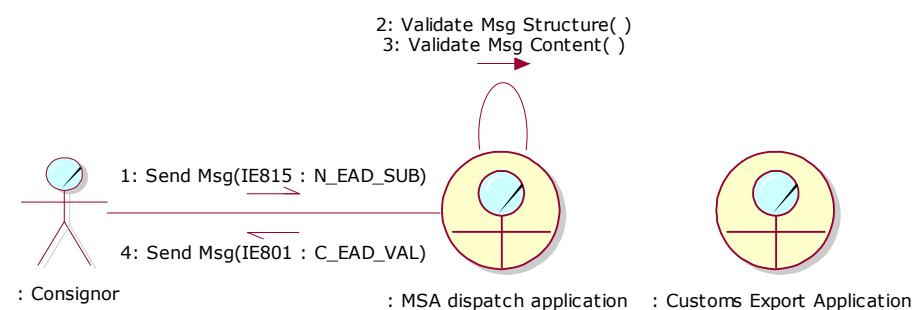


Figure 75: CLD - Local Clearance at Export and movement not released by Customs followed by submission of new export declaration

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2 Export Operation at Office of Export when MSA of Dispatch is MSA of export as well (UC2.43)

In the scenarios described in this section, the Consignor submits only the e-AD, whereas the export declaration is submitted by the consignee at the office of export. In addition, the Member State of export is always the Member State of dispatch.

It shall be noted that:

- The actor *MSA dispatch application* in the Sequence Diagrams is the *MSA dispatch/export application* appearing in FESS [A1]. In this case the consignor and the consignee are in the premises of the same MS and the consignee will be read as forwarding agent.
- For simplification reasons the Sequence Diagrams depict the submission and registration of only one draft e-AD. However, it is possible that an export declaration includes more than one ARC. In this case, all “e-AD” references should be read as “all concerned e-ADs”.
- The scenarios in this section are initiated by the Consignor who submits the draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application with the destination type set to export (Destination Type Code is “6 = Destination - Export”). All the IE messages exchanged in the scenarios of this section containing the ARC and sequence number, namely the IE801, IE818, IE829 and IE839 messages, include the sequence number “1”. It should however be noted that especially the IE839 message contains the sequence number only when the reason for the negative cross-checking (<Rejection Reason Code>) is ‘2: The contents of the e-AD does not match with export data’.

However, the scenarios in this section may also be initiated by the Consignor who submits an update message (IE813: C_UPD_DAT) to the MSA dispatch application with destination export (Destination Type Code is “6 = Destination - Export”) either retaining the destination type code of an export movement or changing the destination type code of a movement from non-export to export. Then, the sequence number of the ARC will be the one indicated in the last IE801 or IE813 (whichever is the latest) message that is created by the MSA of Dispatch and shall be greater than “1”.

III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit

In this scenario, all message validations pass successfully. The export declaration (IE515: E_EXP_DAT) is cross-checked successfully against the e-AD, the movement is released by Customs (IE501: C_AER_SND) and the exit from the Community is confirmed.

III.II.2.1.1 Export Operation at office of export (UC2.43)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “Accepted” and the TIM_EAD timer is initiated.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The forwarding agent submits the export declaration (IE515: E_EXP_DAT) to the Customs Export Application, which in turn forwards it to the MSA dispatch application (the communication between the forwarding agent and the Customs Export Application is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

Upon receipt of the export declaration from the Customs Export Application (IE515: E_EXP_DAT), the MSA dispatch application successfully cross-checks the consistency of the e-AD with the export declaration (IE515: E_EXP_DAT).

In addition, the MSA dispatch application receives the Anticipated Export Record (IE501: C_AER_SND) from the Customs Export Application. Following, the MSA dispatch application:

- Changes the status of the e-AD to “*Exporting*”;
- Builds a notification message (IE829: C_EXP_NOT) and sends it to the Consignor, to the Consignee (forwarding agent) and to the Customs Export Application.

The MSA dispatch application is waiting for the discharge to take place when the exit from the Community is completed.

III.II.2.1.2 Export confirmation of Exit (UC2.46)

The MSA dispatch application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is accepted (IE518/Control Result Code: A1, A2, A4), the MSA dispatch application builds the Report of Export (IE818) reporting exit acceptance (IE818/Global conclusion of receipt: 21, 22) and forwards it to the Consignor. Moreover, the state of the movement at the MSA of Dispatch is updated from “*Exporting*” to “*Delivered*”. Finally, if the TIM_EAD timer has not expired, the MSA dispatch application stops it, otherwise it resets the flag raised locally at its expiration.

The scenario with the acceptance of delivery is depicted in Figure 76 and Figure 77:

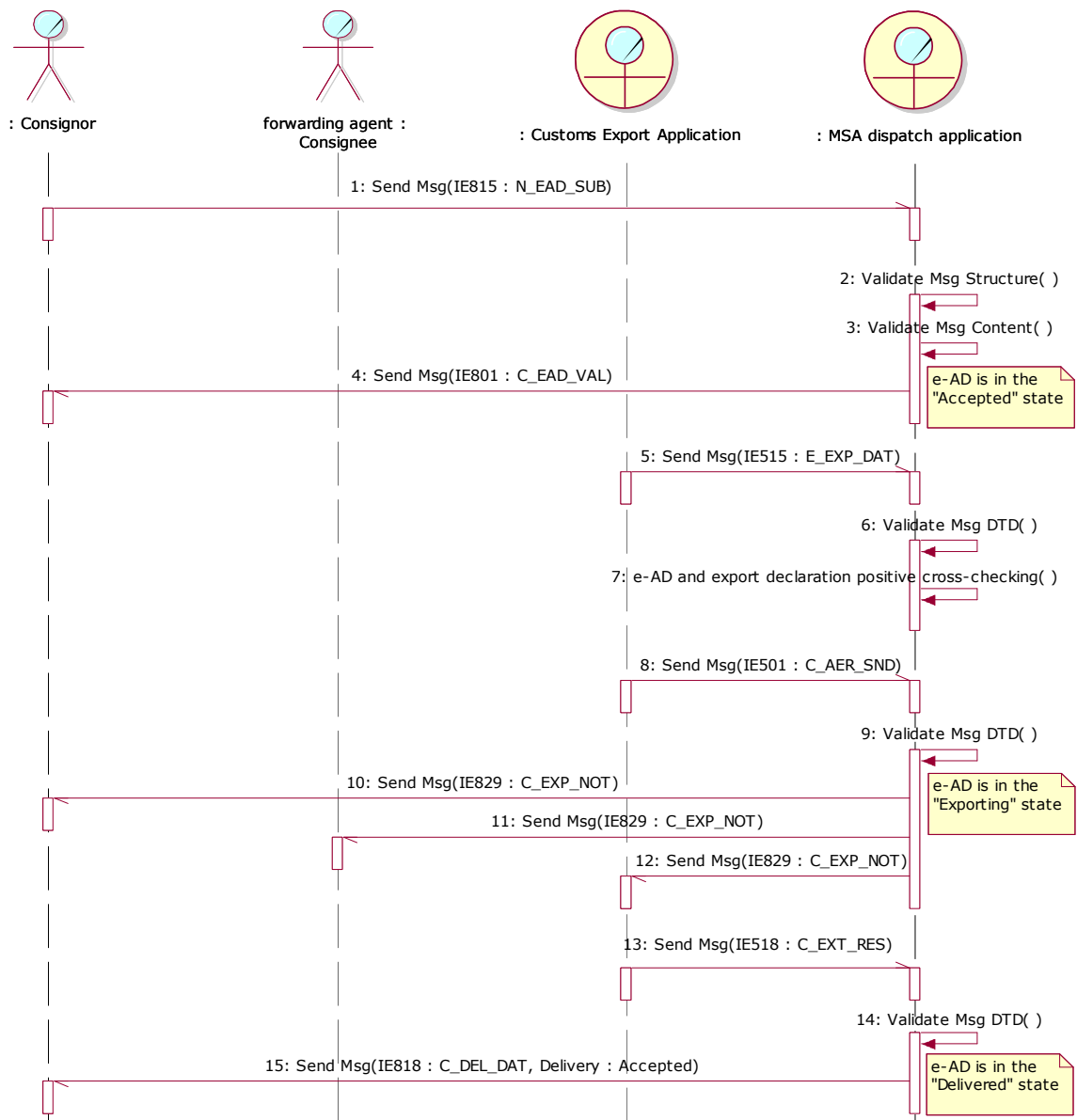


Figure 76: TSD - Export Operation at Office of Export followed by Export confirmation of exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

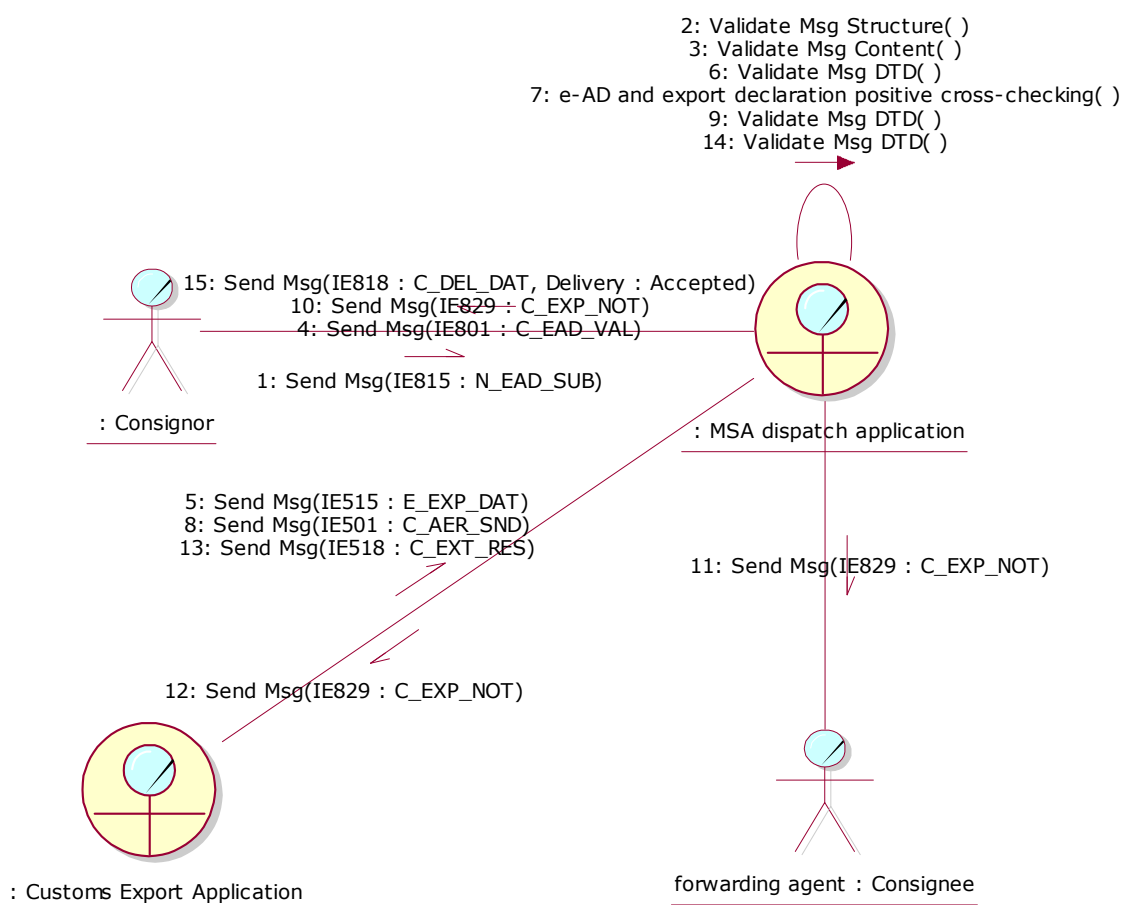


Figure 77: CLD - Export Operation at Office of Export followed by Export confirmation of exit

III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit

In this scenario, all message validations pass successfully. The export declaration (IE515: E_EXP_DAT) is cross-checked successfully against the e-AD, the movement is released by Customs (IE501: C_AER_SND) but the exit is refused.

III.II.2.2.1 Export Operation at office of export (UC2.43)

The “Local Clearance at Export” is exactly the same as in Section III.II.2.1.1 Export Operation at office of export (UC2.43). Hence, the e-AD is in the “Exporting” state and the MSA dispatch application is waiting for the discharge to take place when the exit is completed.

III.II.2.2.2 Export Cancellation of Exit (UC2.46)

The MSA dispatch application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is refused (IE518/Control Result Code: B1), the MSA dispatch application builds the Report of Export (IE818) reporting exit refusal (IE818/Global conclusion of receipt: 23) and forwards it to the Consignor. Moreover, the state of the movement at the MSA of Dispatch is updated from “Exporting” to “Refused”. Finally, it starts the TIM_CHS timer waiting for the change of destination from the Consignor.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2.2.3 Change of Destination (UC2.05)

Upon receipt of the exit refusal (IE818/Global conclusion of receipt: 23), the Consignor issues a change of destination, as described in the scenario of Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 78 and Figure 79:

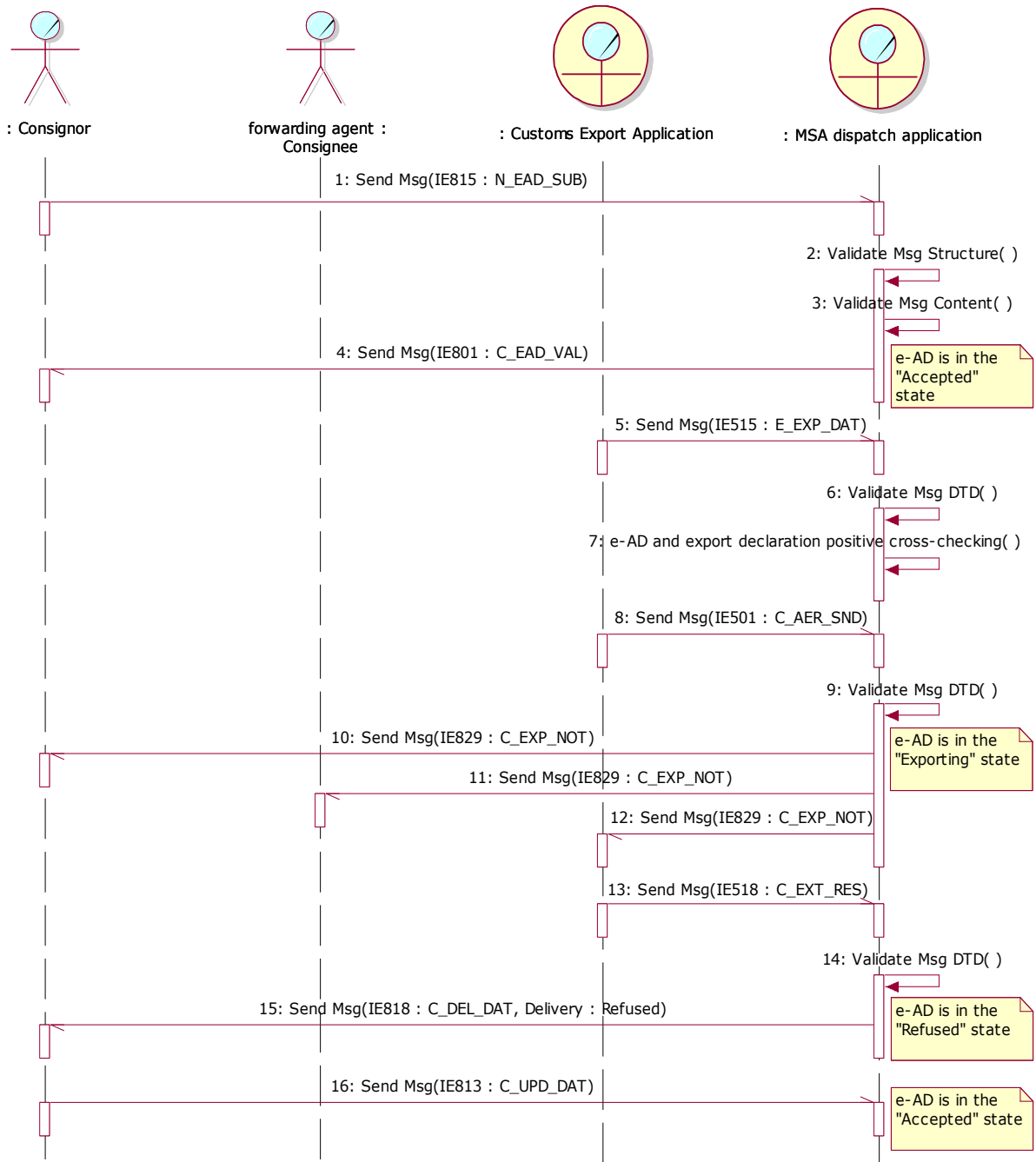


Figure 78: TSD - Export Operation at Office of Export followed by Export cancellation of exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

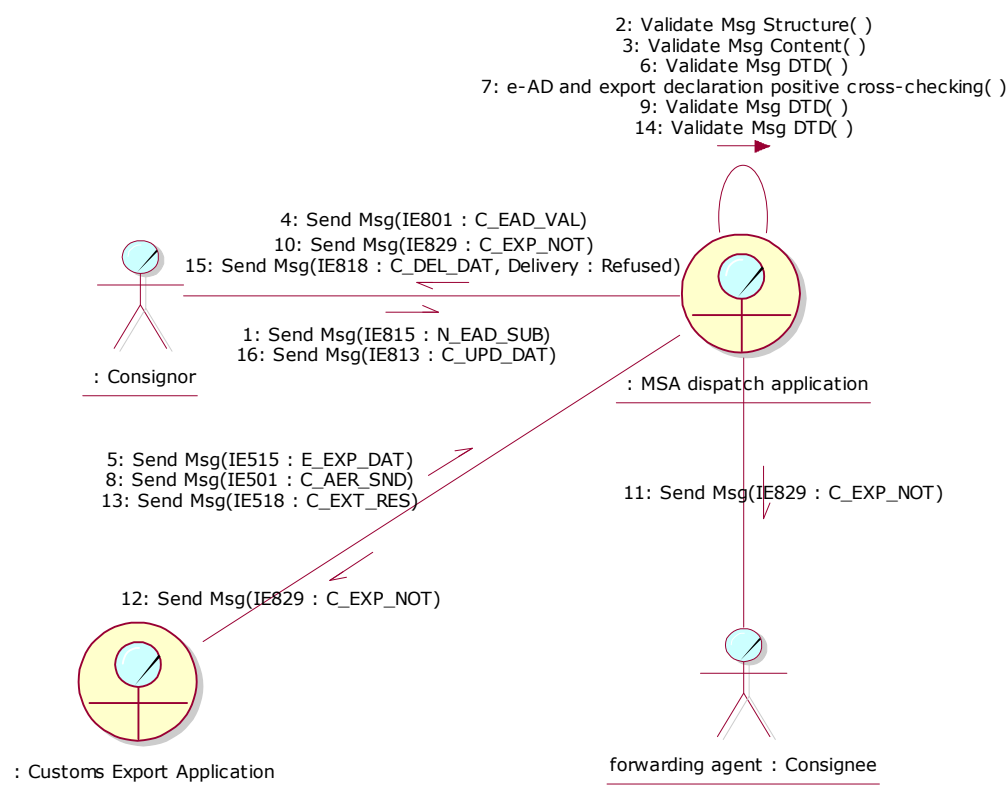


Figure 79: CLD - Export Operation at Office of Export followed by Export cancellation of exit

III.II.2.3 Export Operation at Office of Export when MSA of Dispatch is MSA of export as well, followed by export declaration cancellation and Change of Destination

In this scenario, all message validations pass successfully. The Anticipated Export Record (IE501: C_AER_SND) is cross-checked successfully against the e-AD but the Consignor cancels the export declaration and submits a change of destination.

III.II.2.3.1 Export Operation at office of export (UC2.43)

The “Export Operation at office of export” is exactly the same as in Section III.II.2.1.1 Export Operation at office of export (UC2.43). Hence, the e-AD is in the “Exporting” state, the TIM_EAD has been initiated and the MSA dispatch application is waiting for the discharge to take place when the exit from the Community is completed.

III.II.2.3.2 Change of Destination (UC2.05)

The Consignor cancels the export operation in ECS while the e-AD is in the “Exporting” state. Following the export operation cancellation in ECS, the Consignor submits a change of destination, as described in the scenario of Section III.I.5 Change of Destination (UC2.05).

The change of destination should be performed since the cancellation of the export operation in ECS is not foreseen to be communicated to EMCS. After the issuance of the change of destination by the Consignor, the state of the e-AD in EMCS changes from “Exporting” to “Accepted” at the MSA of Dispatch. In case a change of destination is not submitted by the Consignor after the cancellation of the export operation in ECS, then the e-AD in EMCS will remain in the “Exporting” state at the MSA of Dispatch, the timer TIM_EAD will expire, and the MSA authorities should take action.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The scenario with the export declaration cancelation by the Consignor and the change of destination is depicted in Figure 80 and Figure 81:

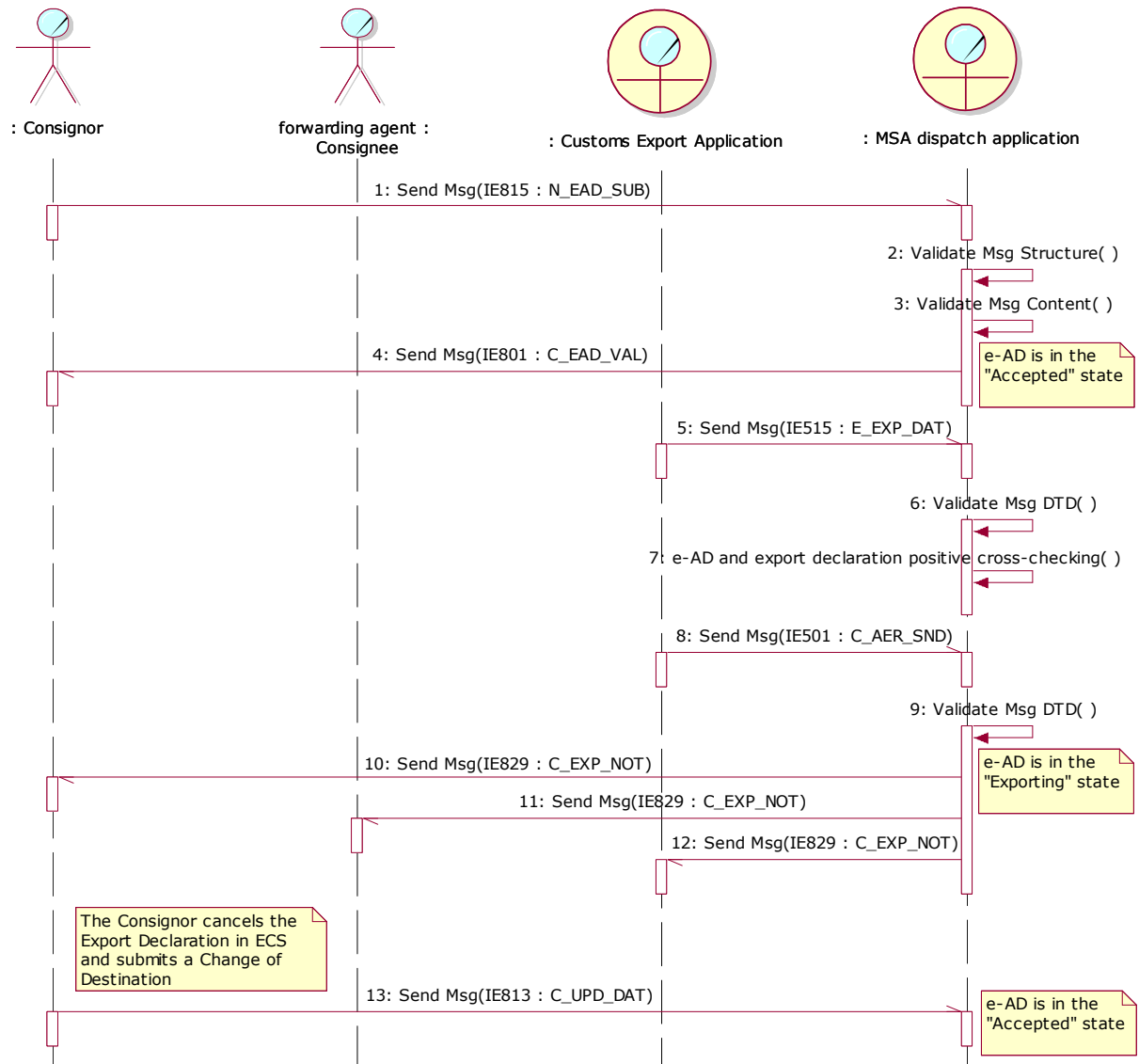


Figure 80: TSD - Export Operation at Office of Export when MSA of Dispatch is MSA of export as well, followed by export declaration cancelation and Change of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

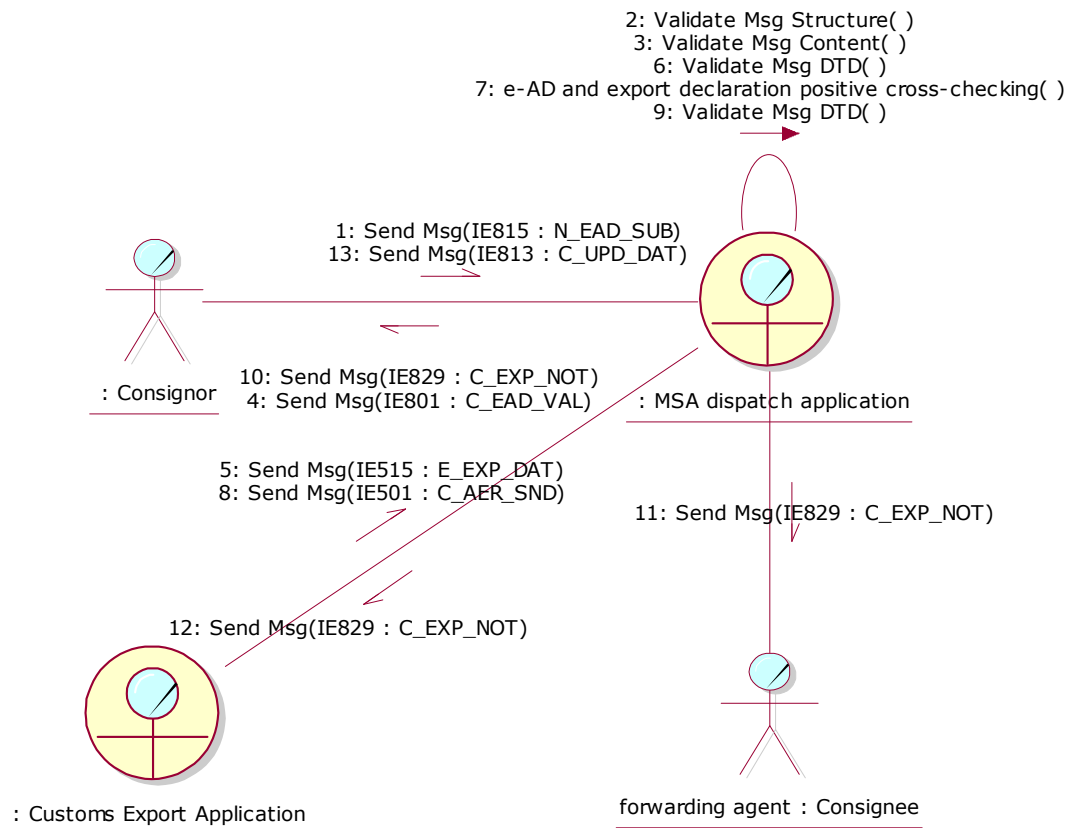


Figure 81: CLD - Export Operation at Office of Export when MSA of Dispatch is MSA of export as well, followed by export declaration cancellation and Change of Destination

III.II.2.4 Export Operation at Office of Export followed by negative cross-checking before the export release and Change of Destination

In this scenario, all message validations pass successfully. However, the cross-checking of the export declaration (IE515: E_EXP_DAT) against the e-AD is found to be negative before the movement is released by Customs.

III.II.2.4.1 Export Operation at office of export (UC2.43)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor. Finally, the state of the movement at the MSA of Dispatch is set to “Accepted” and the TIM_EAD timer is initiated.

The forwarding agent submits the export declaration (IE515: E_EXP_DAT) to the Customs Export Application, which in turn forwards it to the MSA destination application (the communication between the forwarding agent and the Customs Export Application is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

Upon receipt of the export declaration from the Customs Export Application (IE515: E_EXP_DAT), the MSA destination application cross-checks its consistency against the e-AD.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

When the cross-checking is negative, the MSA dispatch application builds a rejection message (IE839: C_CUS_REJ) that includes the list of errors found during the cross-checking of the IE515 with the e-AD and sends it to the Consignor, to the forwarding agent as well as to the Customs Export Application.

Finally, the state of the movement at the MSA of Dispatch is retained to “Accepted” and the TIM_CHS timer is started waiting for the change of destination from the Consignor.

III.II.2.4.2 Change of Destination (UC2.05)

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the Consignor performs a change of destination as described in Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 82 and Figure 83:

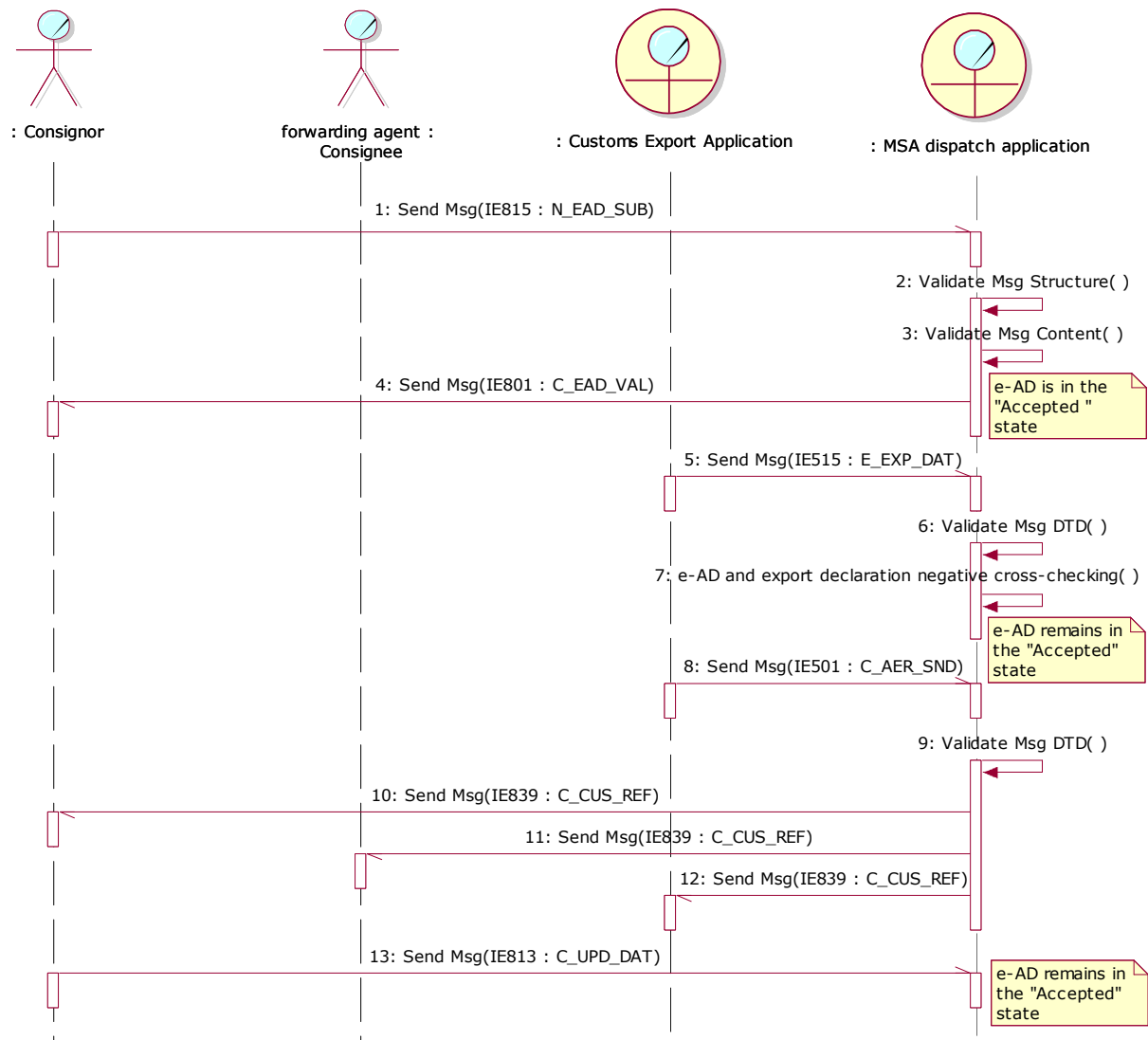


Figure 82: TSD - Export Operation at Office of Export followed by negative cross-checking before the export release and Change of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

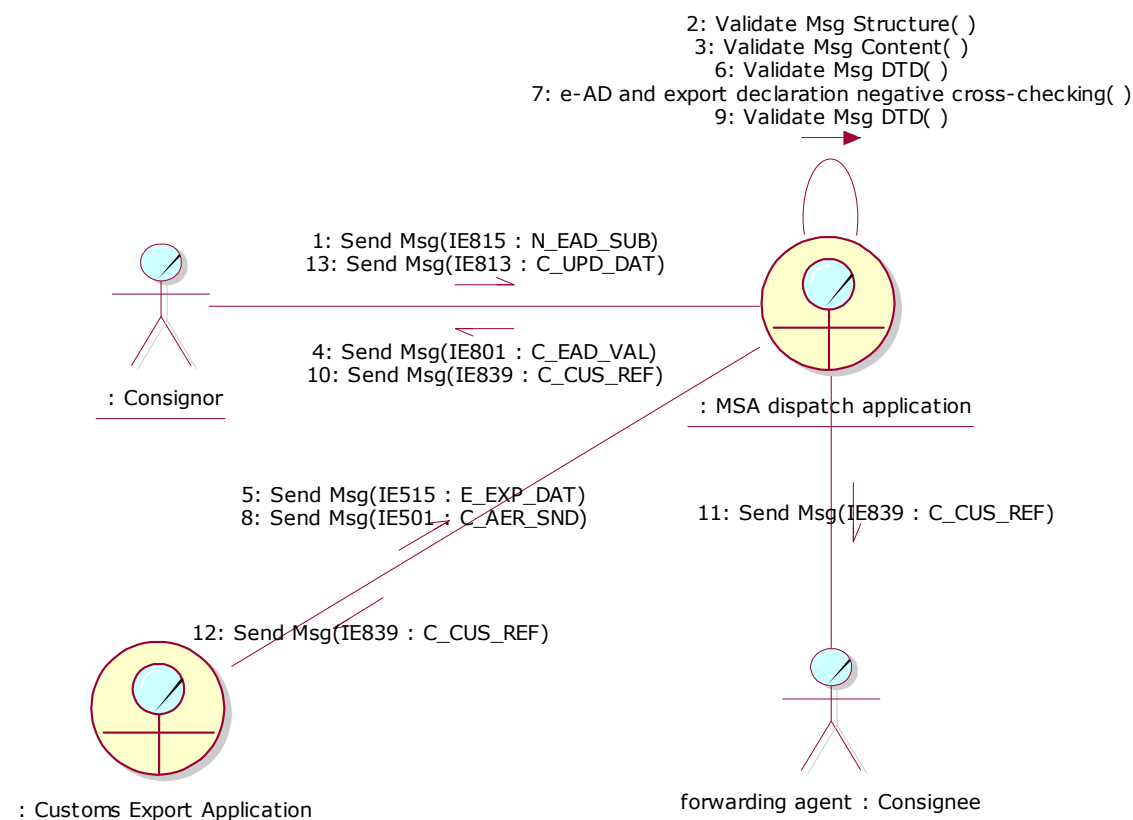


Figure 83: CLD - Export Operation at Office of Export followed by negative cross-checking before the export release and Change of Destination

III.II.2.5 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of new export declaration

In this scenario, all message validations pass successfully. However, the cross-checking of the export declaration (IE515: E_EXP_DAT) against the e-AD is found to be negative even though the movement has already been released by Customs. Following, the forwarding agent cancels the export declaration and submits a new one, which - provided that the movement is released by Customs - is cross-checked against the existing “Accepted” e-AD.

III.II.2.5.1 Export Operation at office of export (UC2.43)

In this scenario, the cross-checking between the export declaration (IE515: E_EXP_DAT) and the e-AD is found to be negative, as described earlier in the scenario of Section III.II.2.4.1 Export Operation at office of export (UC2.43). Hence, the movement is still in the “Accepted” state. However, the movement has already been released by Customs.

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the forwarding agent cancels the export declaration and submits a new one (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

This scenario may be followed by:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- Positive cross-checking, release by Customs and export confirmation of exit (Section III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit);
- Positive cross-checking, release by Customs, and export cancellation of exit (Section III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit);
- Negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.2.4 Export Operation at Office of Export followed by negative cross-checking *before* the export release and Change of Destination, Section III.II.2.5 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of new export declaration, Section III.II.2.6 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of change of destination);
- No release by Customs (Section III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration, Section III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination).

This scenario is depicted in Figure 84 and Figure 85:

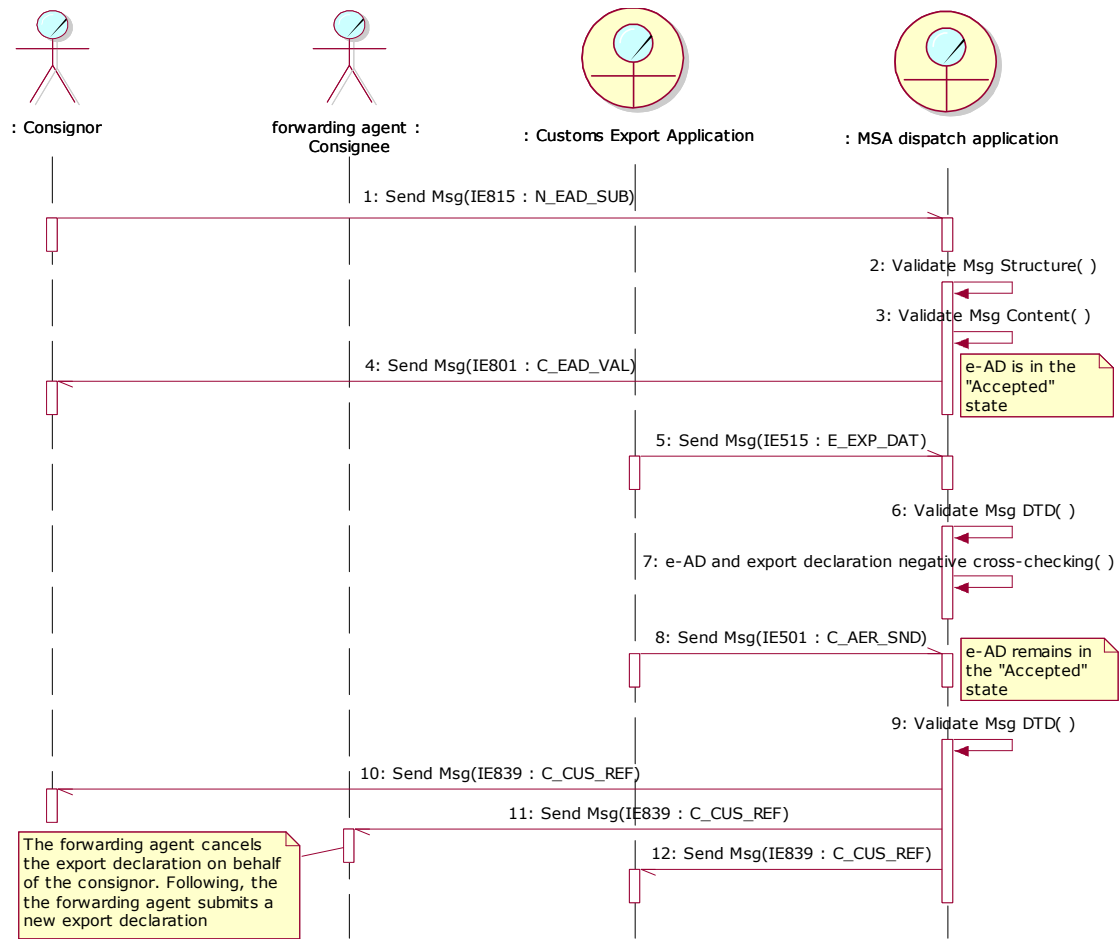


Figure 84: TSD - Export Operation at Office of Export followed by negative cross-checking after the export release, export declaration cancellation and submission of new export declaration

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

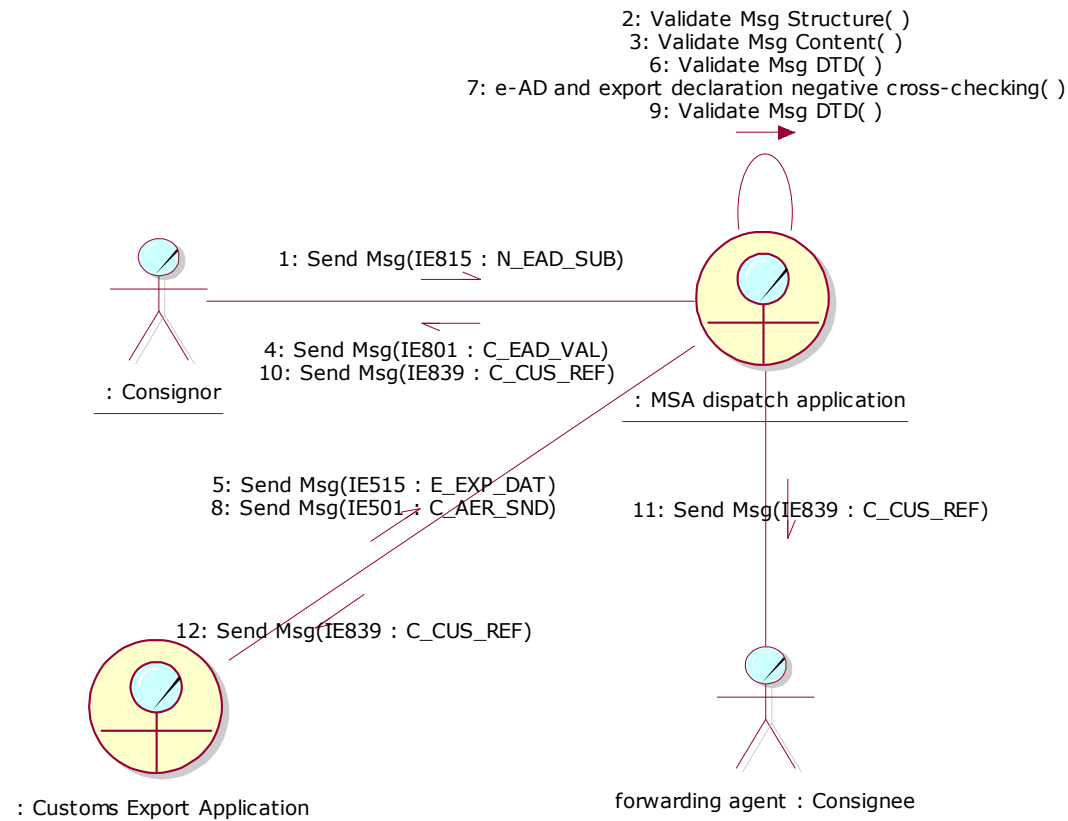


Figure 85: CLD - Export Operation at Office of Export followed by negative cross-checking after the export release, export declaration cancellation and submission of new export declaration

III.II.2.6 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of change of destination

In this scenario, all message validations pass successfully. However, the cross-checking of the export declaration (IE515: E_EXP_DAT) against the e-AD is found to be negative even though the movement has already been released by Customs. Following, the forwarding agent cancels the export declaration and issues a change of destination for the “Accepted” e-AD.

III.II.2.6.1 Export Operation at office of export (UC2.43)

In this scenario, the cross-checking between the export declaration (IE515: E_EXP_DAT) and the e-AD is found to be negative, as described earlier in the scenario of Section III.II.2.4.1 Export Operation at office of export (UC2.43). Hence, the movement is still in the “Accepted” state. However, the movement has already been released by Customs.

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the forwarding agent cancels the export declaration (this communication is outside EMCS, hence it is not displayed in the relevant Sequence and Collaboration Diagrams).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2.6.2 Change of Destination (UC2.05)

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the Consignor performs a change of destination as described in Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 86 and Figure 87:

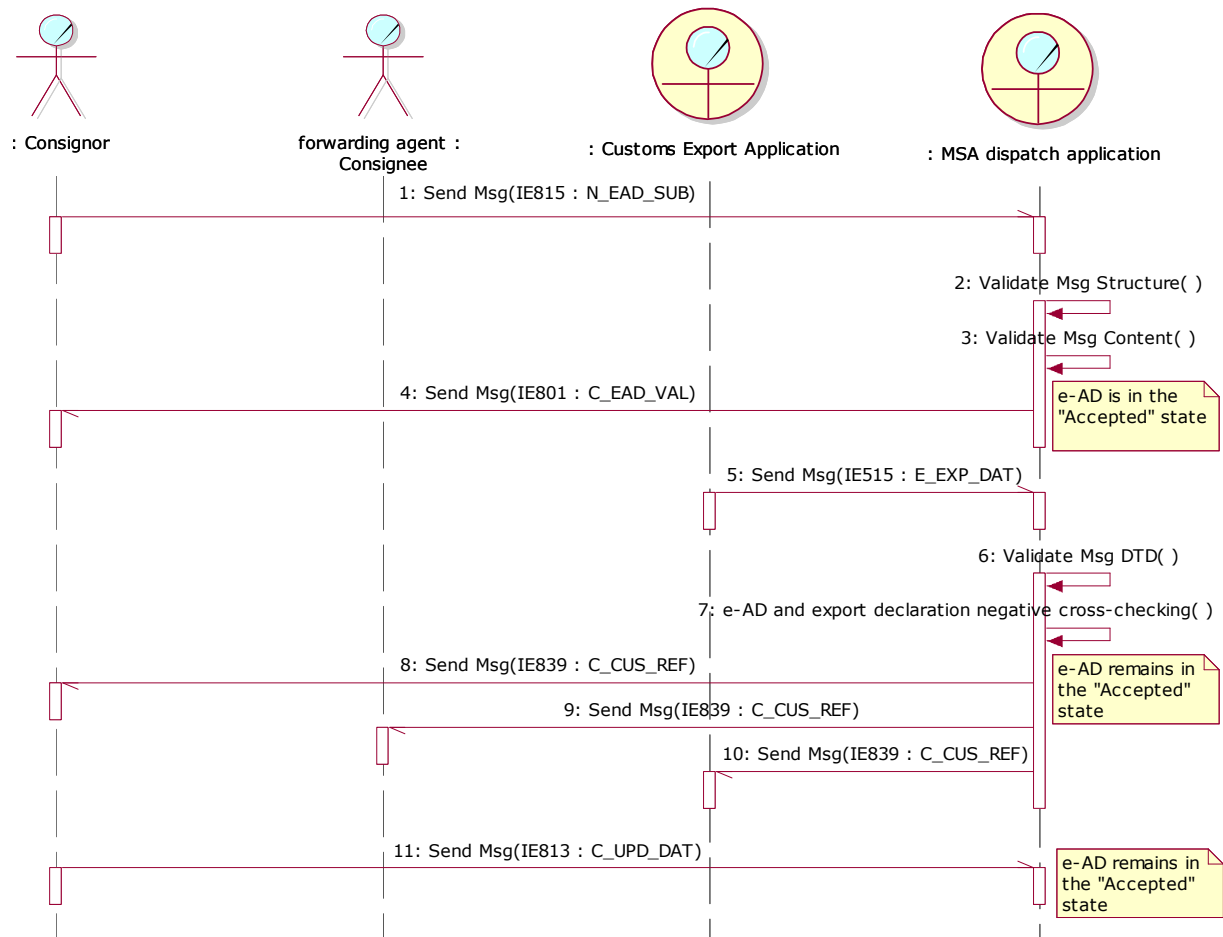


Figure 86: TSD - Export Operation at Office of Export followed by negative cross-checking after the export release, export declaration cancellation and submission of change of destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

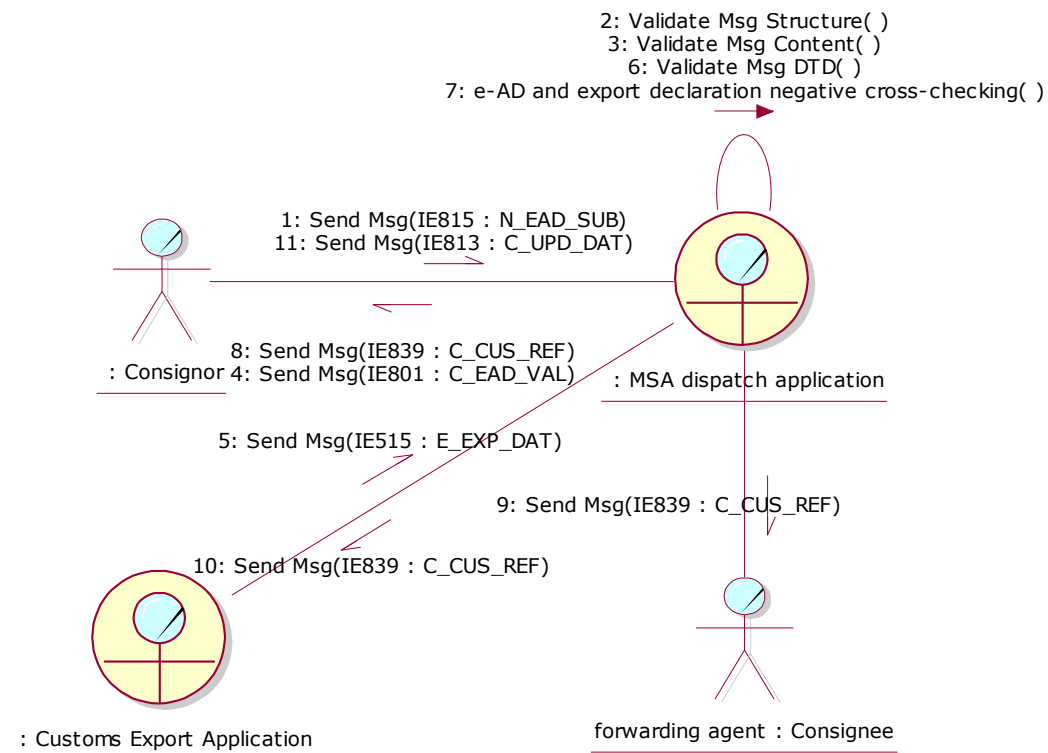


Figure 87: CLD - Export Operation at Office of Export followed by negative cross-checking after the export release, export declaration cancellation and submission of change of destination

III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration

In this scenario, the movement has not been released by Customs, and therefore the e-AD has been maintained in the “Accepted” state. Following, the forwarding agent submits a new export declaration.

This scenario may be followed by:

- Positive cross-checking, release by Customs and export confirmation of exit (Section III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit);
- Positive cross-checking, release by Customs, and export cancellation of exit (Section III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit);
- Negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.2.4 Export Operation at Office of Export followed by negative cross-checking *before* the export release and Change of Destination, Section III.II.2.5 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of new export declaration, Section III.II.2.6 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of change of destination);

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- No release by Customs (Section III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration, Section III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination).

This scenario is depicted in Figure 88 and Figure 89:

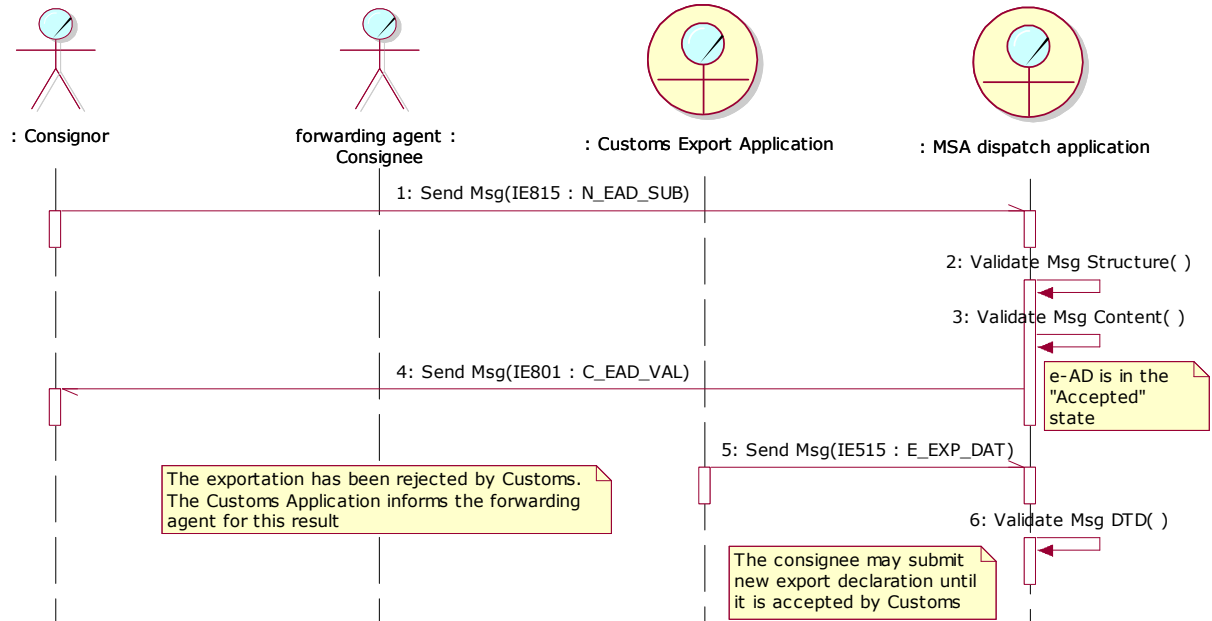


Figure 88: TSD - Export Operation at Office of Export and movement not released by Customs followed by new export declaration

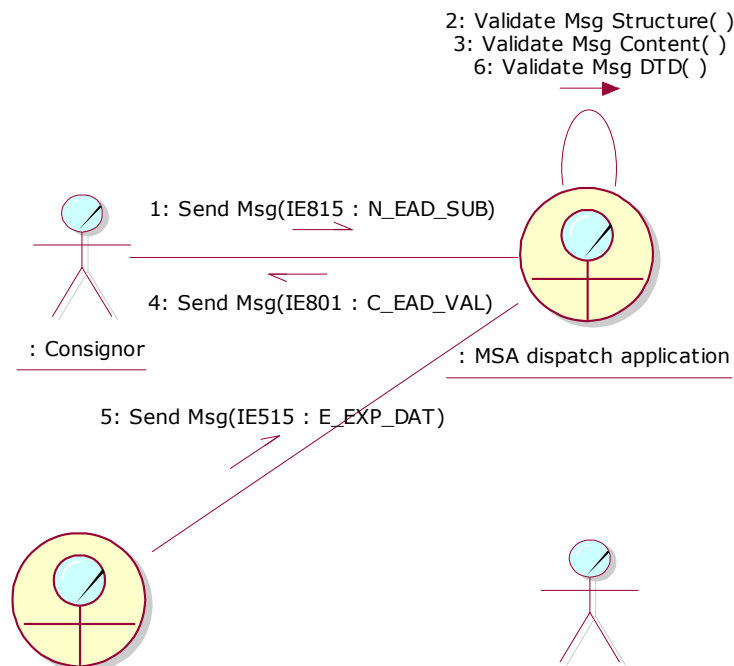


Figure 89: CLD - Export Operation at Office of Export and movement not released by Customs followed by new export declaration

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination

In this scenario, the movement has not been released by Customs, and therefore the e-AD has been maintained in the "Accepted" state. Following, the Consignor issues a change of destination.

This scenario is depicted in Figure 90 and Figure 91:

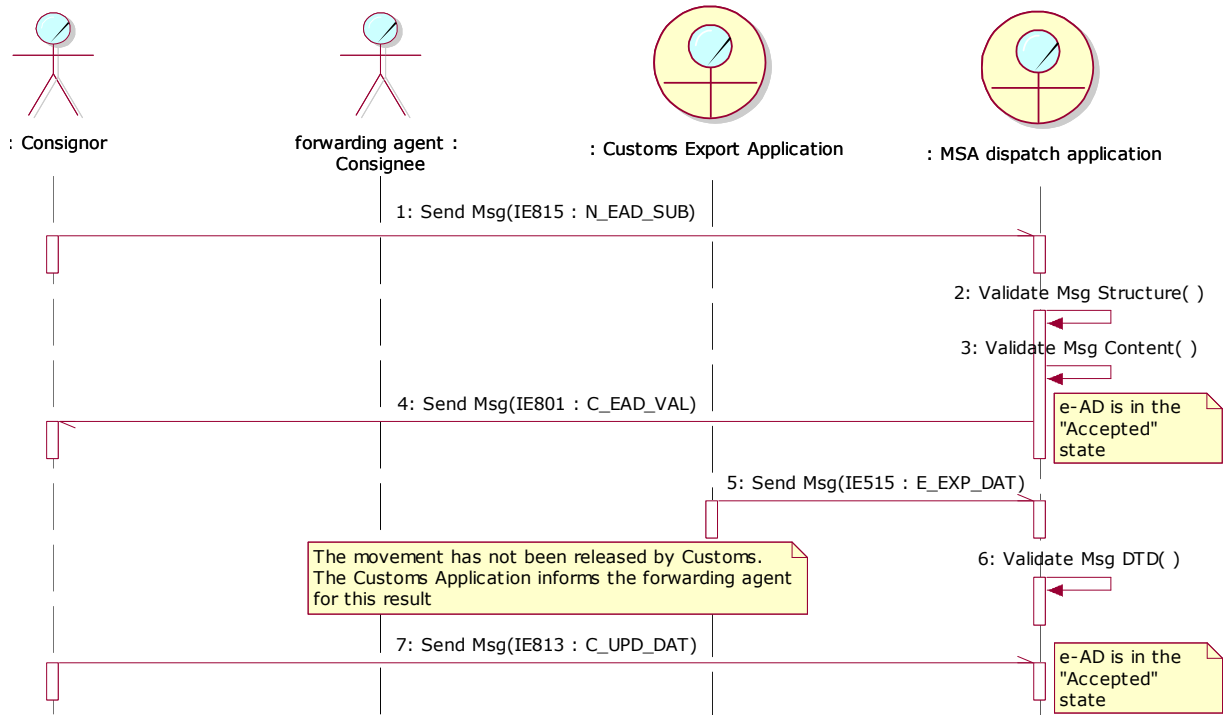


Figure 90: TSD - Export Operation at Office of Export and movement not released by Customs followed by change of destination

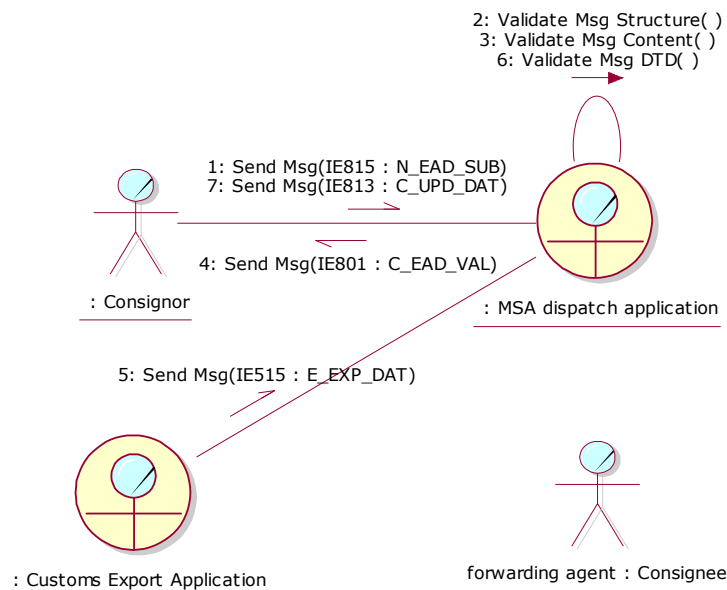


Figure 91: CLD - Export Operation at Office of Export and movement not released by Customs followed by change of destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.3 Export Operation at Office of Export when MSA of Dispatch is different from MSA of export (UC2.43)

Just like in Section III.II.2 Export Operation at Office of Export when MSA of Dispatch is MSA of export as well (UC2.43), in the scenarios described in this section, the Consignor submits only the e-AD, whereas the export declaration is submitted by the consignee at the office of export. The only difference is that the Member State of export is different than Member State of dispatch.

It shall be noted that:

- The actor MSA destination application in the Sequence Diagrams is the MSA export application appearing in FESS [A1].
- For simplification reasons the Sequence Diagrams depict the submission and registration of only one draft e-AD. However, it is possible that an export declaration includes more than one ARC. In this case, all “e-AD” references should be read as “all concerned e-ADs”.
- The scenarios in this section are initiated by the Consignor who submits the draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application with the destination type set to export (Destination Type Code is “6 = Destination - Export”). All the IE messages exchanged in the scenarios of this section containing the ARC and sequence number, namely the IE801, IE818, IE829 and IE839 messages, include the sequence number “1”. It should however be noted that especially the IE839 message contains the sequence number only when the reason for the negative cross-checking (<Rejection Reason Code>) is ‘2: The contents of the e-AD does not match with export data’.

However, the scenarios in this section may also be initiated by the Consignor who submits an update message (IE813: C_UPD_DAT) to the MSA dispatch application with destination export (Destination Type Code is “6 = Destination - Export”) either retaining the destination type code of an export movement or changing the destination type code of a movement from non-export to export. Then, the sequence number of the ARC will be the one indicated in the last IE801 or IE813 (whichever is the latest) message that is created by the MSA of Dispatch and received by the MSA of Export and shall be greater than “1”.

III.II.3.1 Export Operation at Office of Export followed by Export Confirmation of Exit (UC2.43)

In this scenario, all validations pass successfully. The export declaration (IE515: E_EXP_DAT) is cross-checked successfully against the e-AD, the movement is released by Customs (IE501: C_AER_SND) and the exit is confirmed.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.3.1.1 Export Operation at office of export (UC2.43)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). Upon successful validation of the draft e-AD, the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor and also disseminates it to the MSA destination application. Finally, the state of the movement at the MSA of Dispatch is set to “*Accepted*” and the TIM_EAD timer is initiated.

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the MSA destination application stores the e-AD and sets the state of the e-AD at MSA of Destination to “*Accepted*”. Finally, the MSA destination application forwards the e-AD (IE801: C_EAD_VAL) to the forwarding agent.

The forwarding agent submits the export declaration (IE515: E_EXP_DAT) to the Customs Export Application, which in turn forwards it to the MSA destination application. Upon receipt of the export declaration from the Customs Export Application (IE515: E_EXP_DAT), the MSA destination application cross-checks successfully the consistency of the e-AD with the export declaration (IE515: E_EXP_DAT).

In addition, the MSA destination application receives the Anticipated Export Record (IE501: C_AER_SND) from the Customs Export Application. Following, the MSA destination application:

- Changes the status of the e-AD to “*Exporting*”;
- Builds a notification message (IE829: C_EXP_NOT) and sends it to the forwarding agent, to the Customs Export Application and to the MSA dispatch application.

Upon receipt of the notification message (IE829: C_EXP_NOT), the MSA dispatch application performs the following actions:

- Changes the status of the e-AD to “*Exporting*”;
- Sends the notification message (IE829: C_EXP_NOT) to the Consignor.

The MSA dispatch application and the MSA destination application are waiting for the discharge to take place when the exit from the Community is completed.

III.II.3.1.2 Export confirmation of Exit (UC2.46)

The MSA destination application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is accepted (IE518/Control Result Code: A1, A2, A4), the MSA destination application builds the Report of Export (IE818) reporting exit acceptance (IE818/Global conclusion of receipt: 21, 22) and forwards it to the MSA dispatch application. Moreover, the state of the movement at the MSA of Destination is updated from “*Exporting*” to “*Delivered*”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon the reception of the delivery notification message (IE818/Global conclusion of receipt: 21, 22, 23), the MSA dispatch application validates it successfully and changes the state of the e-AD from "Exporting" to "Delivered". Moreover, if the TIM_EAD timer has not expired, the MSA dispatch application stops it otherwise it resets the flag raised locally at its expiration. In addition, the MSA dispatch application forwards the delivery notification (IE818: C_DEL_DAT) to the Consignor.

This scenario is depicted in Figure 92 and Figure 93:

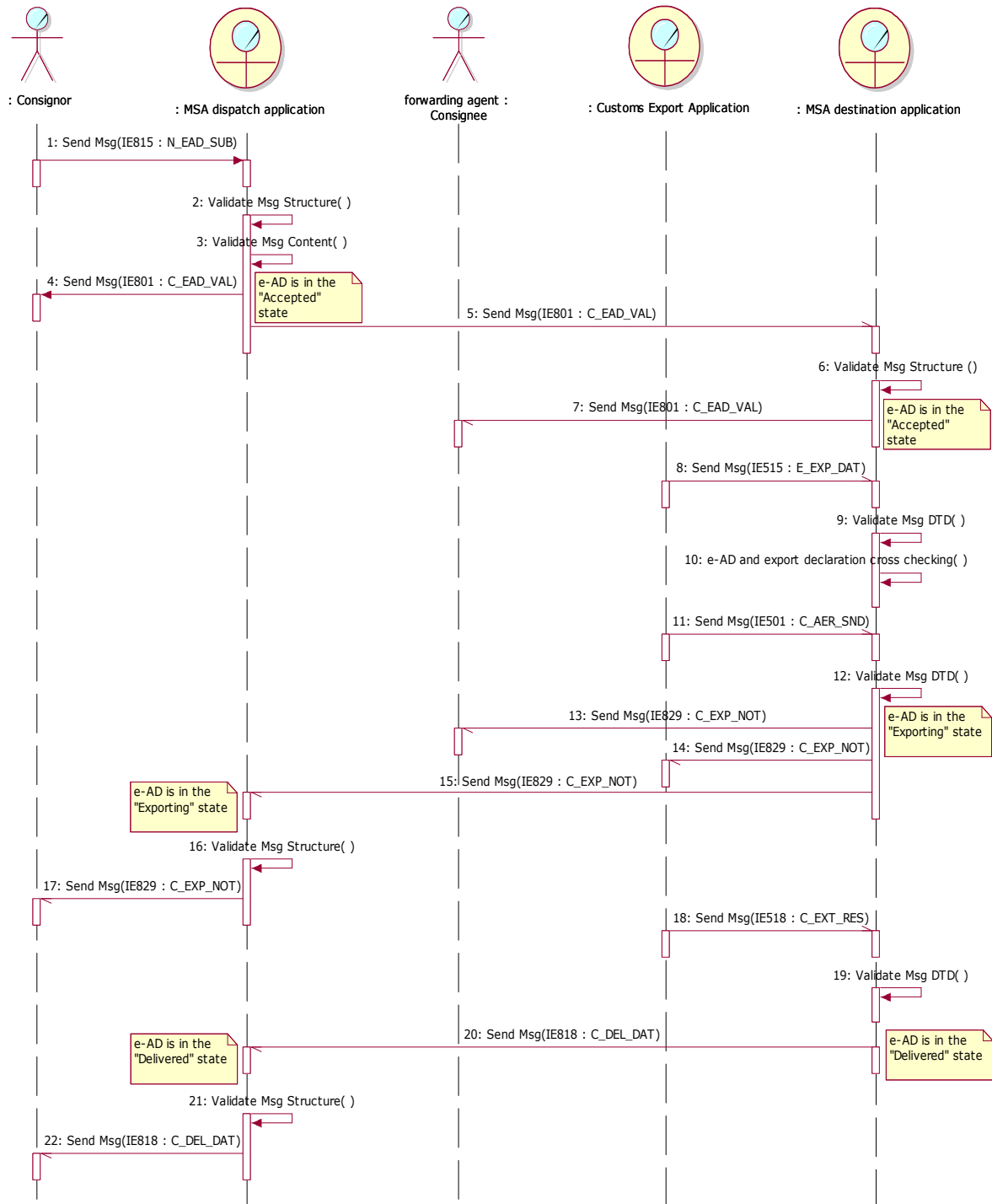


Figure 92: TSD - Export Operation at Office of Export followed by Export confirmation of exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

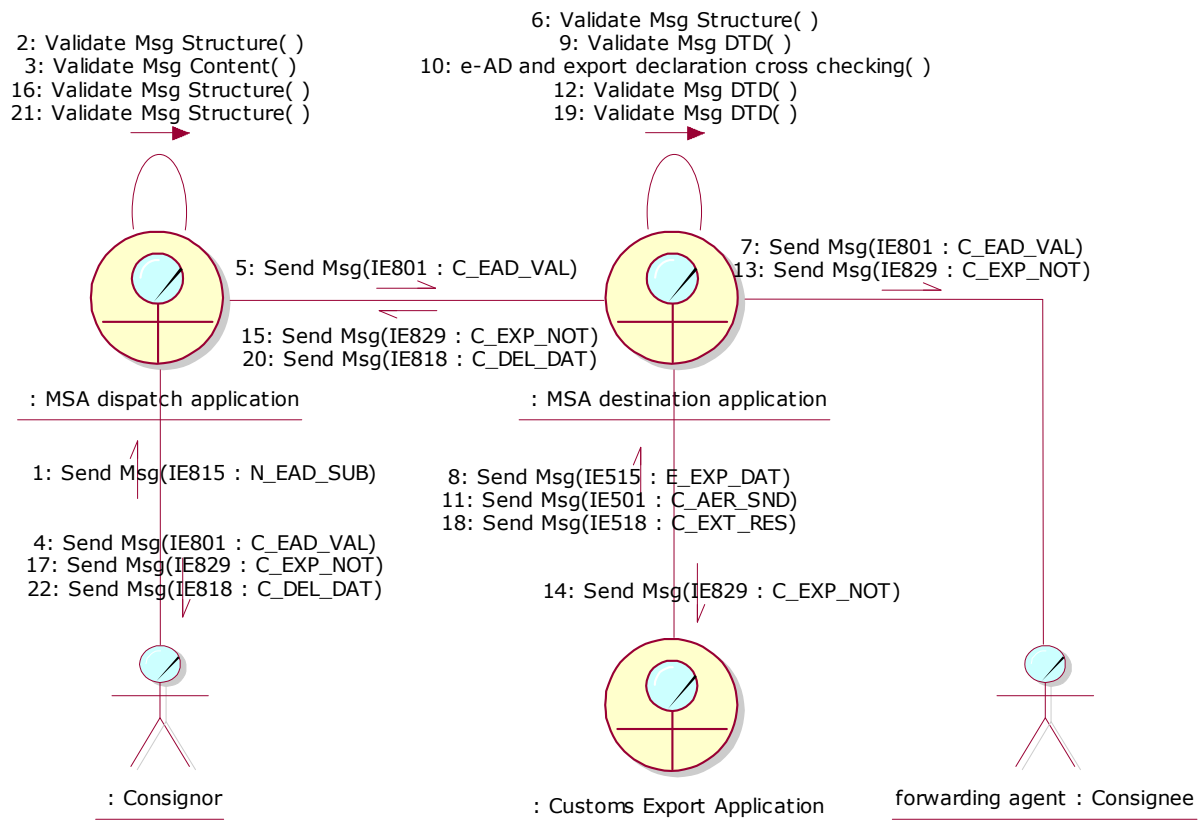


Figure 93: CLD - Export Operation at Office of Export followed by Export confirmation of exit

III.II.3.2 Export Operation at Office of Export followed by Export Cancellation of exit

In this scenario, all validations pass successfully. The export declaration (IE515: E_EXP_DAT) is cross-checked successfully against the e-AD, the movement is released by Customs (IE501: C_AER_SND) but the exit is refused.

III.II.3.2.1 Export Operation at office of export (UC2.43)

This Section is exactly the same as Section III.II.3.1.1. Hence, the e-AD is in the “Exporting” state and the MSA dispatch application is waiting for the discharge to take place when the exit is completed.

III.II.3.2.2 Export Cancellation of Exit (UC2.46)

The MSA destination application receives the exit results from the Customs Export Application (IE518: C_EXT_RES). Assuming that the validation process passes successfully and the exit is refused (IE518/Control Result Code: B1), the MSA destination application builds the Report of Export (IE818) reporting exit refusal (IE818/Global conclusion of receipt: 23) and forwards it to MSA dispatch application. Moreover, the state of the movement at the MSA of Dispatch is updated from “Exporting” to “Refused”.

Upon the reception of the delivery notification message (IE818/Global conclusion of receipt: 23), the MSA dispatch application validates it successfully and changes the state of the e-AD from “Exporting” to “Refused”. In addition, the MSA dispatch application forwards the delivery notification (IE818: C_DEL_DAT) to the Consignor.

Finally, it starts the TIM_CHS timer waiting for the change of destination from the Consignor as described in Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 94 and Figure 95:

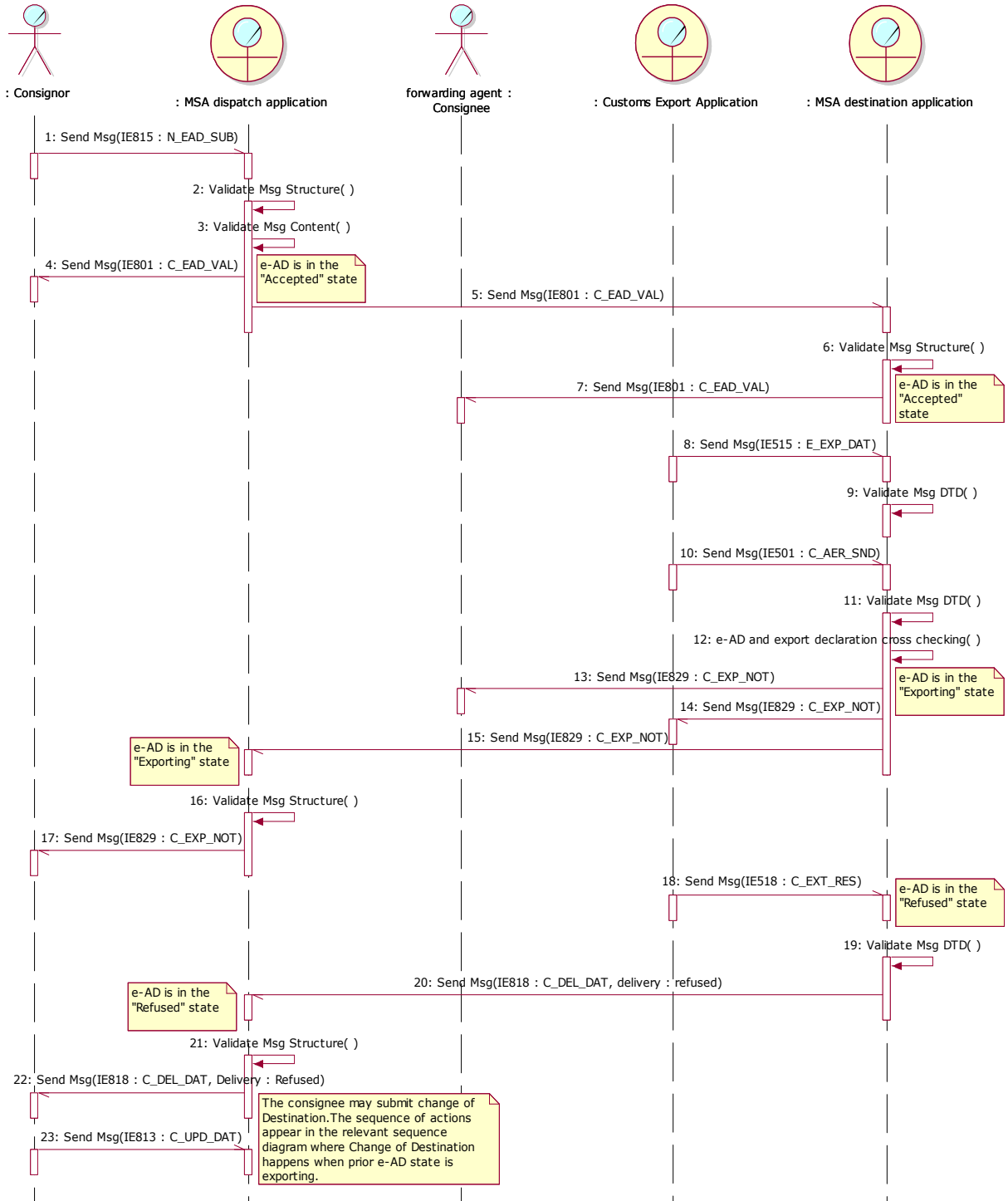


Figure 94: TSD - Export Operation at Office of Export followed by Export cancellation of exit

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

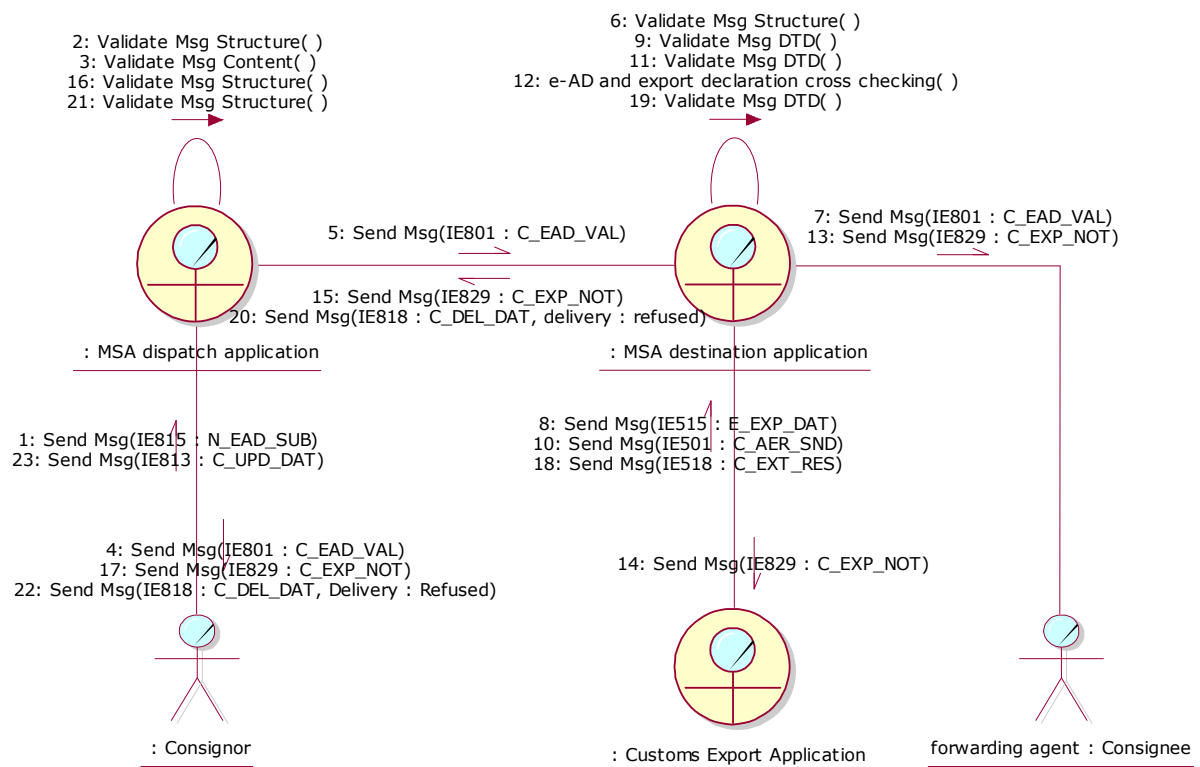


Figure 95: CLD - Export Operation at Office of Export followed by Export cancellation of exit

III.II.3.3 Export Operation at Office of Export followed by cross-checking failure *before* the export release and Change of Destination

In this scenario the e-AD is validated properly by the MSA dispatch application and the export declaration is forwarded to the MSA destination application but the crosschecking fails before the export would probably be released by the Customs Export Application.

III.II.3.3.1 Export Operation at office of export (UC2.43)

According to this scenario, the Consignor submits a draft e-AD (IE815: N_EAD_SUB) to the MSA dispatch application including export as the destination type (Destination Type Code is “6 = Destination - Export”). The validation of the submitted draft e-AD passes successfully and the MSA dispatch application returns the validated e-AD (IE801: C_EAD_VAL) back to the Consignor and the MSA destination application. Finally, the state of the movement at the MSA of Dispatch is set to “Accepted” and the TIM_EAD timer is initiated.

Upon the reception of the validated e-AD (IE801: C_EAD_VAL) from the MSA dispatch application, the MSA destination application stores the e-AD and sets the state of the e-AD at MSA of Destination to “Accepted”. Finally, the MSA destination application forwards the e-AD (IE801: C_EAD_VAL) to the forwarding agent.

The forwarding agent submits the export declaration (IE515: E_EXP_DAT) to the Customs Export Application, which in turn forwards it to the MSA destination application. Upon receipt of the export declaration from the Customs Export Application (IE515: E_EXP_DAT), the MSA destination application cross-checks the consistency of the e-AD(s) with the export declaration (IE515: E_EXP_DAT).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

When the cross-checking is negative, the MSA destination application builds a rejection message (IE839: C_CUS_REJ) that includes the list of errors found during the cross-checking of the IE515 with the e-AD and sends it to the Consignor, to the Consignee (forwarding agent) as well as to the Customs Export Application.

Finally, the state of the movement at the MSA of Destination and Dispatch is retained to "Accepted" and the TIM_CHS timer is started at dispatch waiting for the change of destination from the Consignor.

III.II.3.3.2 Change of Destination (UC2.05)

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the Consignor performs a change of destination as described in Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 96 and Figure 97:

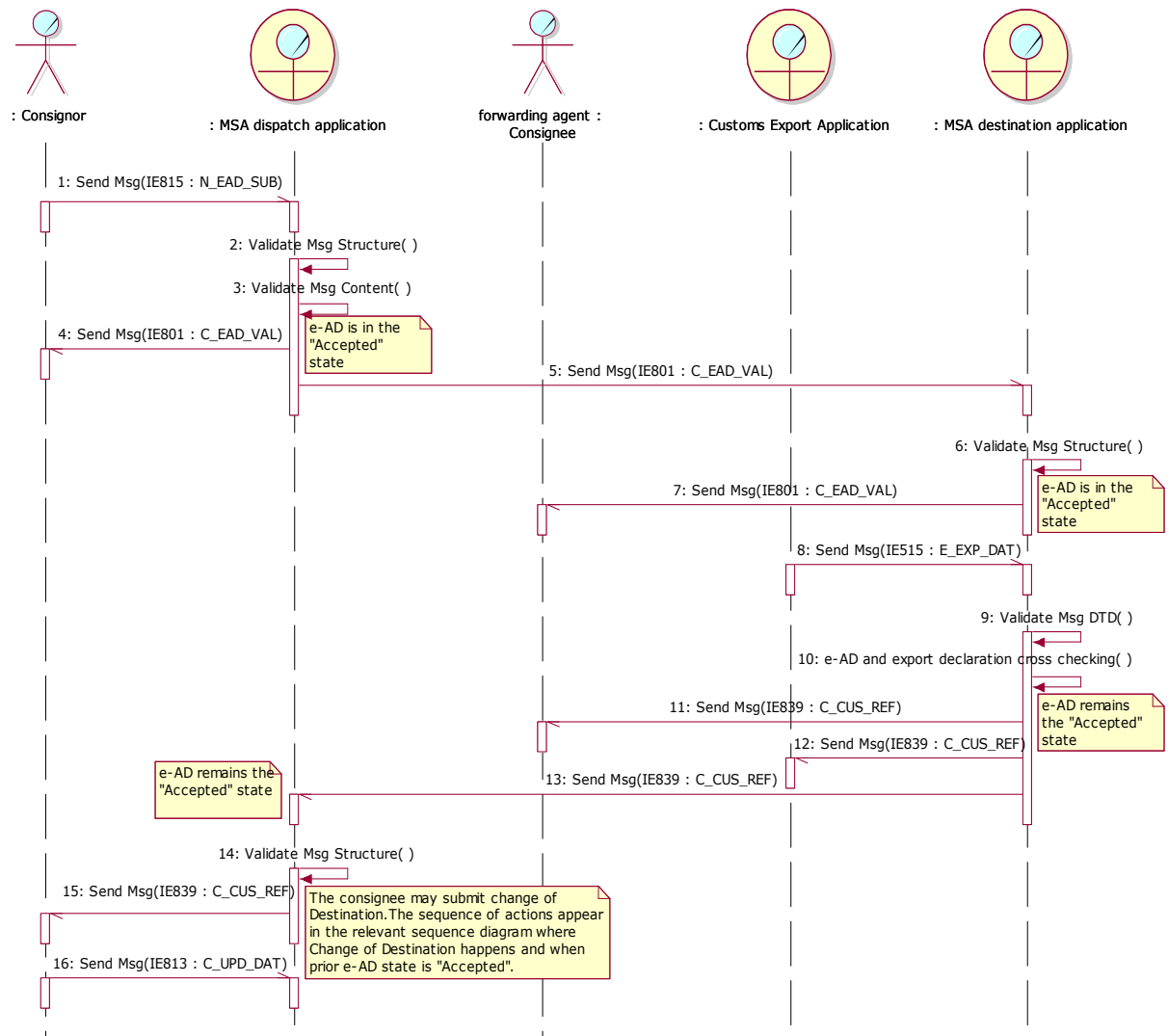


Figure 96: TSD - Export Operation at Office of Export followed by cross-checking failure before the export release and Change of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

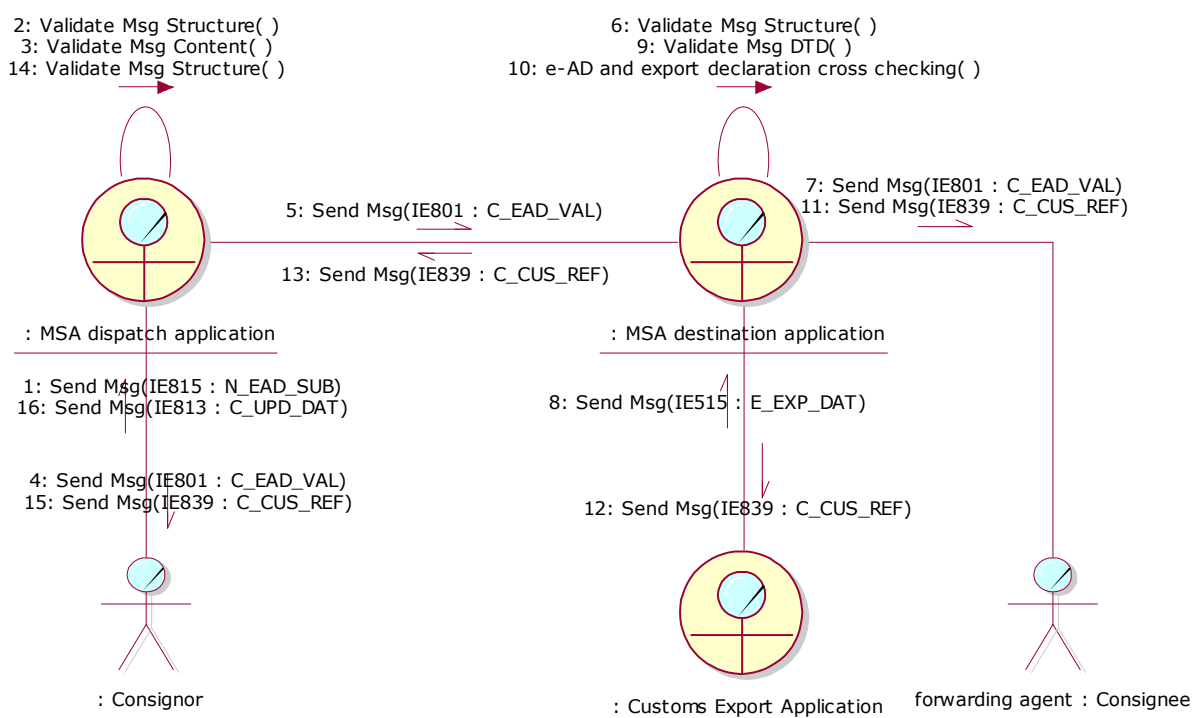


Figure 97: CLD - Export Operation at Office of Export followed by cross-checking failure before the export release and Change of Destination

III.II.3.4 Export Operation at Office of Export when MSA of Dispatch is different from MSA of export, followed by export declaration cancellation and Change of Destination

In this scenario, all validations pass successfully. Upon receipt of the notification message (IE829: C_EXP_NOT), the MSA dispatch application performs the following actions:

- Changes the status of the e-AD to “Exporting”;
- Sends the notification message (IE829: C_EXP_NOT) to the Consignor.

The MSA dispatch application and the MSA destination application are waiting for the discharge to take place but the Consignor cancels the export declaration and submits a change of destination.

III.II.3.4.1 Export Operation at office of export (UC2.43)

The “Export Operation at office of export” is exactly the same as in Section III.II.3.1.1 Export Operation at office of export (UC2.43). Hence, the e-AD is in the “Exporting” state, the TIM_EAD has been initiated and the MSA dispatch application and the MSA destination application are waiting for the discharge to take place when the exit from the Community is completed.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.II.3.4.2 Change of Destination (UC2.05)

Consignor cancels the export operation in ECS while the e-AD is in the “Exporting” state both at the MSA of Dispatch and the MSA of Destination. Following the export operation cancelation in ECS, the Consignor submits a change of destination, as described in the scenario of Section III.I.5 Change of Destination (UC2.05).

The change of destination should be performed since the cancelation of the export operation in ECS is not foreseen to be communicated to EMCS. After the issuance of the change of destination by the Consignor, the state of the e-AD in EMCS changes from “Exporting” to “Accepted” at the MSA of Dispatch. In case a change of destination is not submitted by the Consignor after the cancelation of the export operation in ECS, then the e-AD in EMCS will remain in the “Exporting” state at the MSA of Dispatch, the timer TIM_EAD will expire, and the MSA authorities should take action.

The scenario with the export declaration cancelation by the Consignor and the change of destination is depicted in Figure 98 and Figure 99:

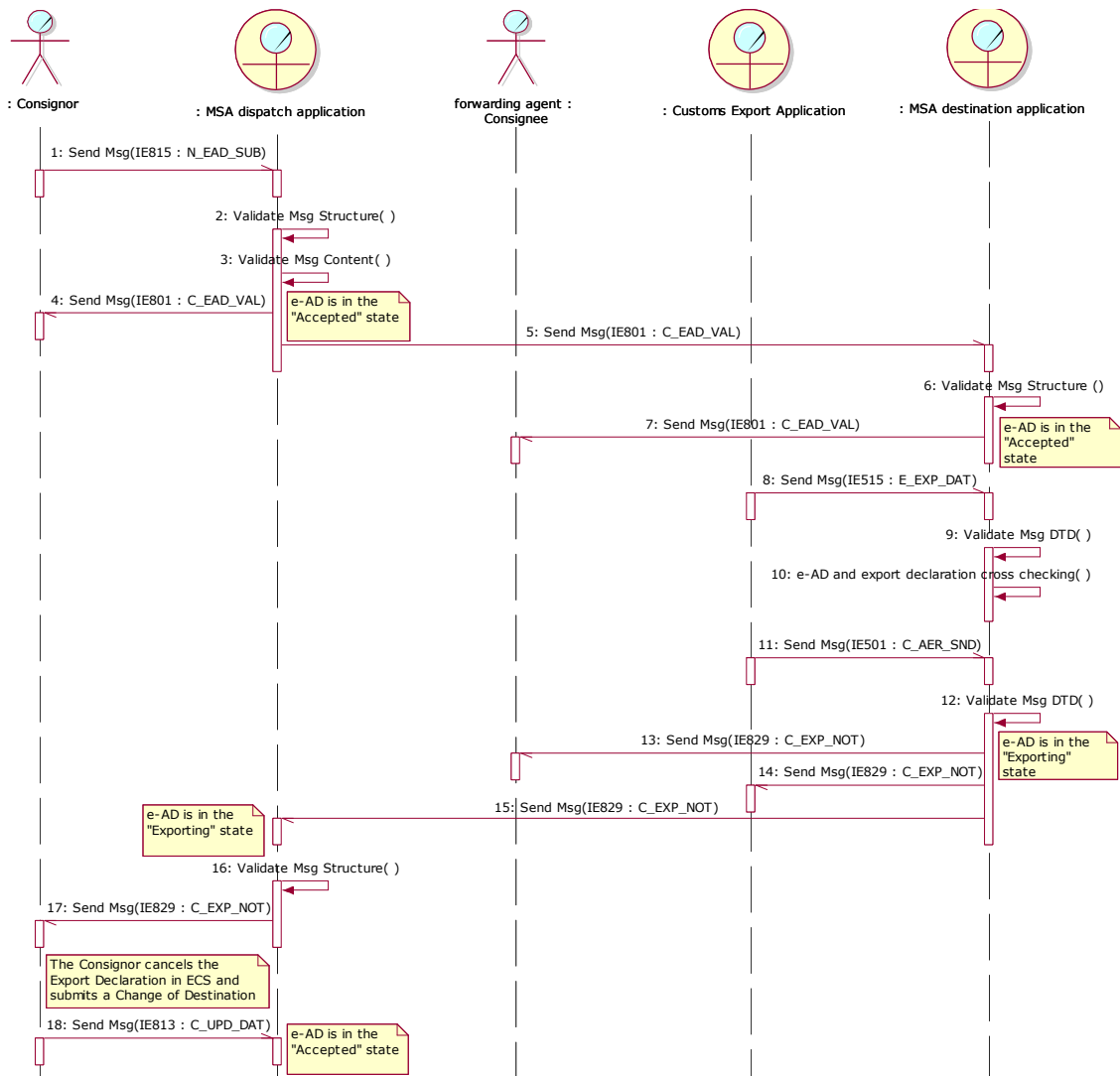


Figure 98: TSD - Export Operation at Office of Export when MSA of Dispatch is different from MSA of export, followed by export declaration cancelation and Change of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

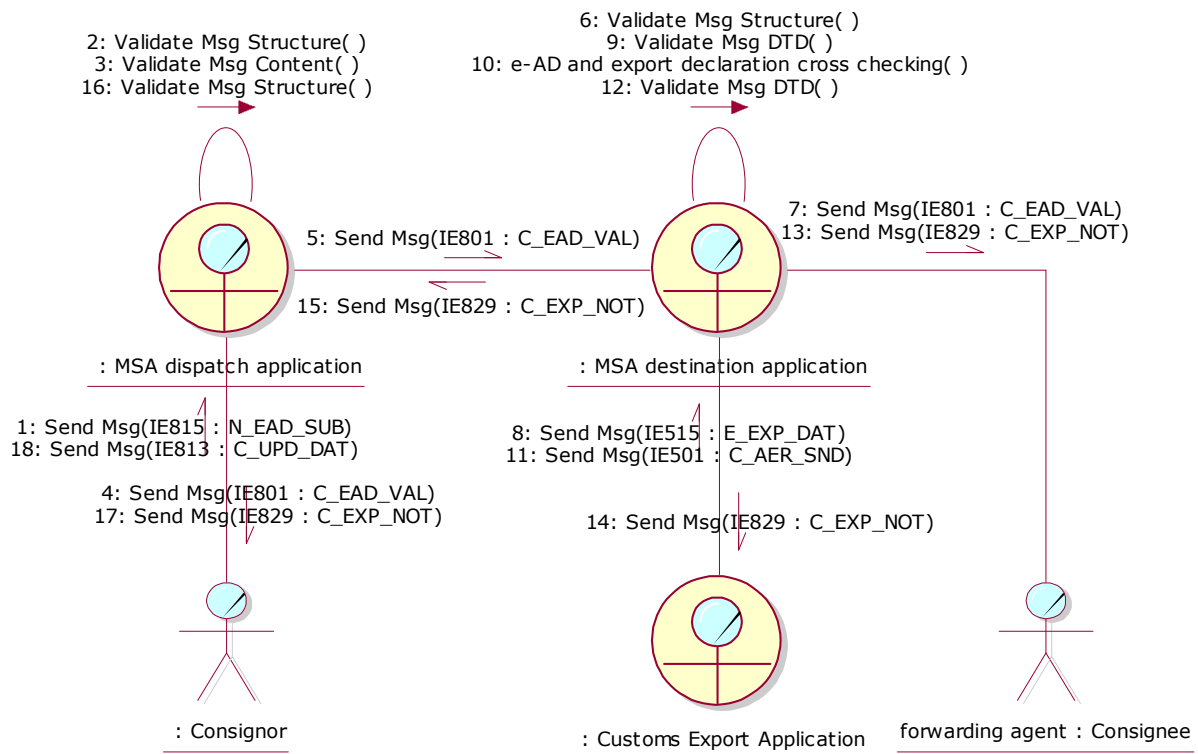


Figure 99: CLD - Export Operation at Office of Export when MSA of Dispatch is different from MSA of export, followed by export declaration cancellation and Change of Destination

III.II.3.5 Export Operation at Office of Export followed by cross-checking failure *after* the export release, export declaration cancellation and submission of new export declaration

In this scenario the e-AD is validated properly by the MSA dispatch application and the export declaration is forwarded to the MSA destination application. However, the cross-checking fails even though the movement has already been released by Customs. Following, the forwarding agent cancels the export declaration and submits a new one, which - provided that the movement is released by Customs - is cross-checked against the existing “Accepted” e-ADs.

III.II.3.5.1 Export Operation at office of export (UC2.43)

In this scenario, the cross-checking between the export declaration (IE515: E_EXP_DAT) and the e-AD has failed as described earlier in the scenario of Section III.II.2.5.1 Export Operation at office of export (UC2.43). Hence, the movement is in the “Accepted” state. The only difference is that the movement has already been released by Customs.

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the forwarding agent cancels the export declaration and submits a new one (through the appropriate ECS messages).

This scenario may be followed by:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- Positive cross-checking, release by Customs and export confirmation of exit (Section III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit);
- Positive cross-checking, release by Customs, and export cancellation of exit (Section III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit);
- Negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.2.4 Export Operation at Office of Export followed by negative cross-checking *before* the export release and Change of Destination, Section III.II.2.5 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of new export declaration, Section III.II.2.6 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of change of destination);
- No release by Customs (Section III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration, Section III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination).

This scenario is depicted in Figure 100 and Figure 101:

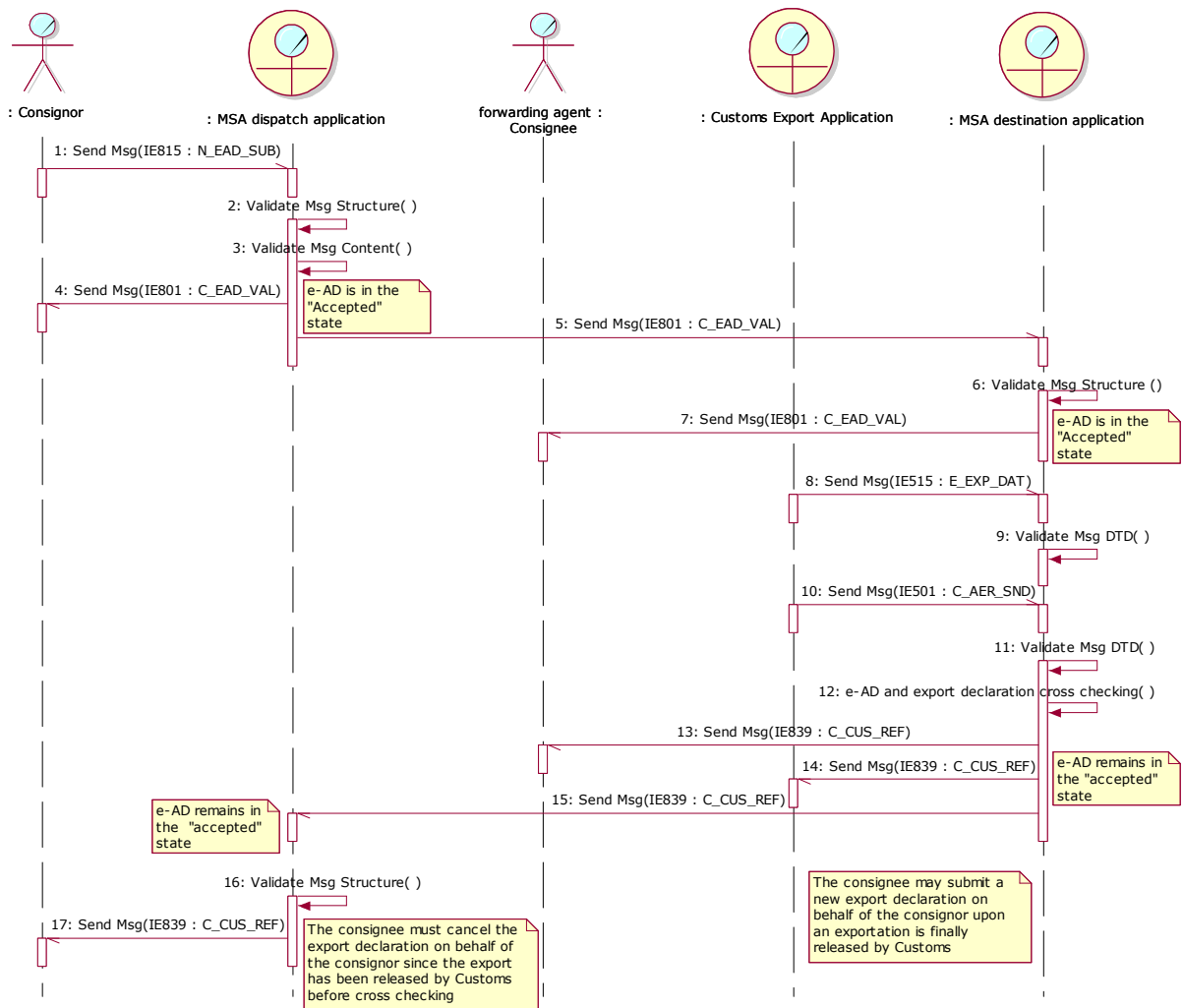


Figure 100: TSD - Export Operation at Office of Export followed by cross-checking failure after the export release followed by Export declaration cancellation and resubmission of new export declaration

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

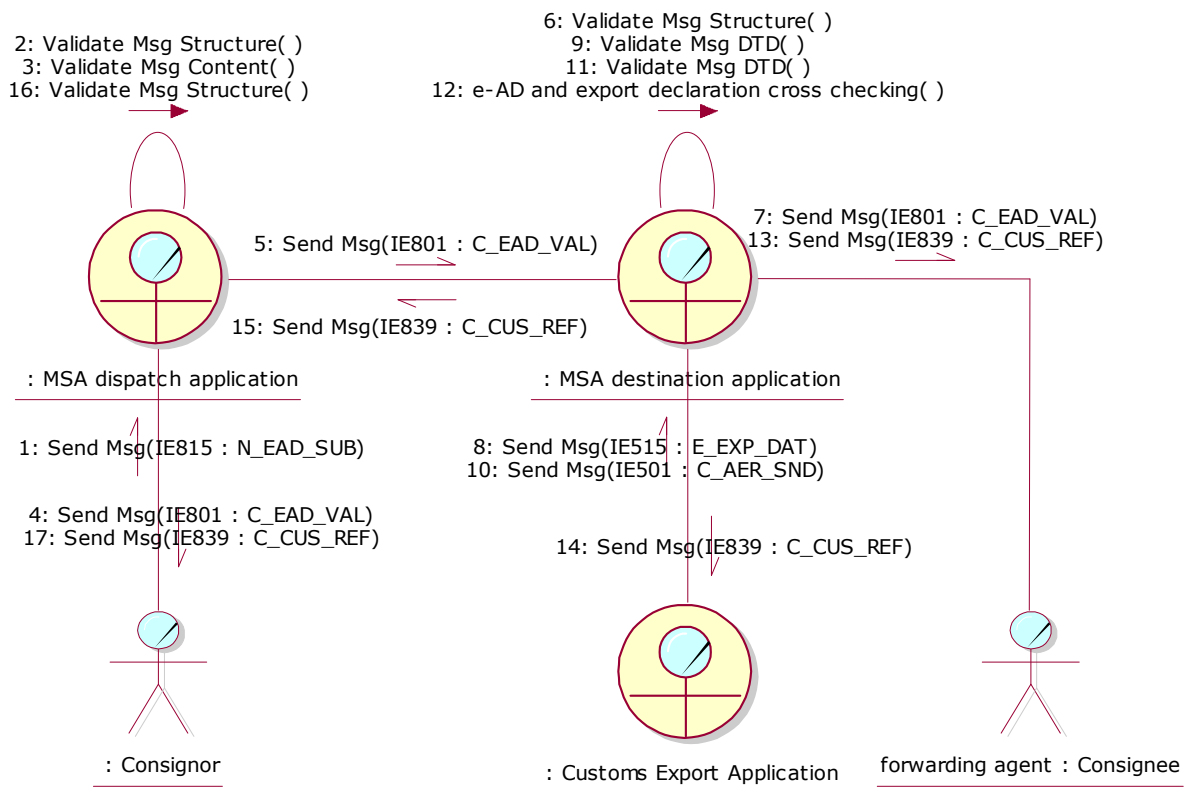


Figure 101: CLD - Export Operation at Office of Export followed by cross-checking failure after the export release followed by Export declaration cancellation and resubmission of new export declaration

III.II.3.6 Export Operation at Office of Export followed by cross-checking failure *after* the export release, export declaration cancellation and submission of change of destination

In this scenario, the e-AD is validated properly by the MSA dispatch application and the export declaration is forwarded to the MSA destination application and to the consignor. However, the cross-checking fails even though the movement has already been released by Customs. Following, the forwarding agent cancels the export declaration, while the Consignor issues a change of destination for the “Accepted” e-AD.

III.II.3.6.1 Export Operation at office of export (UC2.43)

In this scenario, the cross-checking between the export declaration (IE515: E_EXP_DAT) and the e-AD has failed as described earlier in the scenario of Section III.II.2.5.1 Export Operation at office of export (UC2.43). Hence, the movement is in the “Accepted” state. The only difference is that the movement has already been released by Customs.

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the forwarding agent cancels the export declaration.

III.II.3.6.2 Change of Destination (UC2.05)

Upon reception of the rejection message (IE839: C_CUS_REJ) and the examination of the rejection results, the Consignor performs a change of destination as described in Section III.I.5 Change of Destination (UC2.05).

This scenario is depicted in Figure 102 and Figure 103:

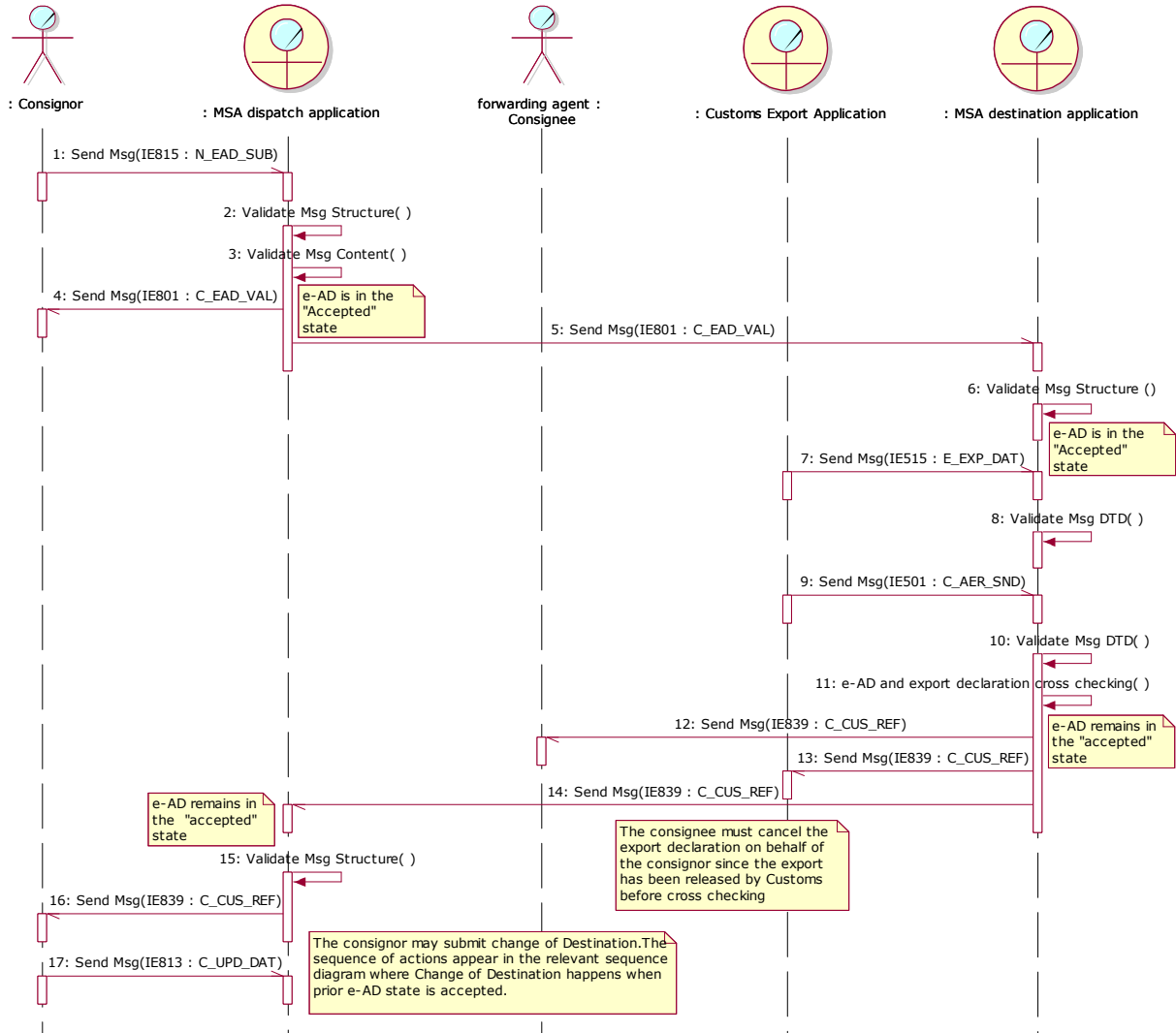


Figure 102: TSD - Export Operation at Office of Export followed by cross-checking failure after the export release, export declaration cancellation and submission of change of destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

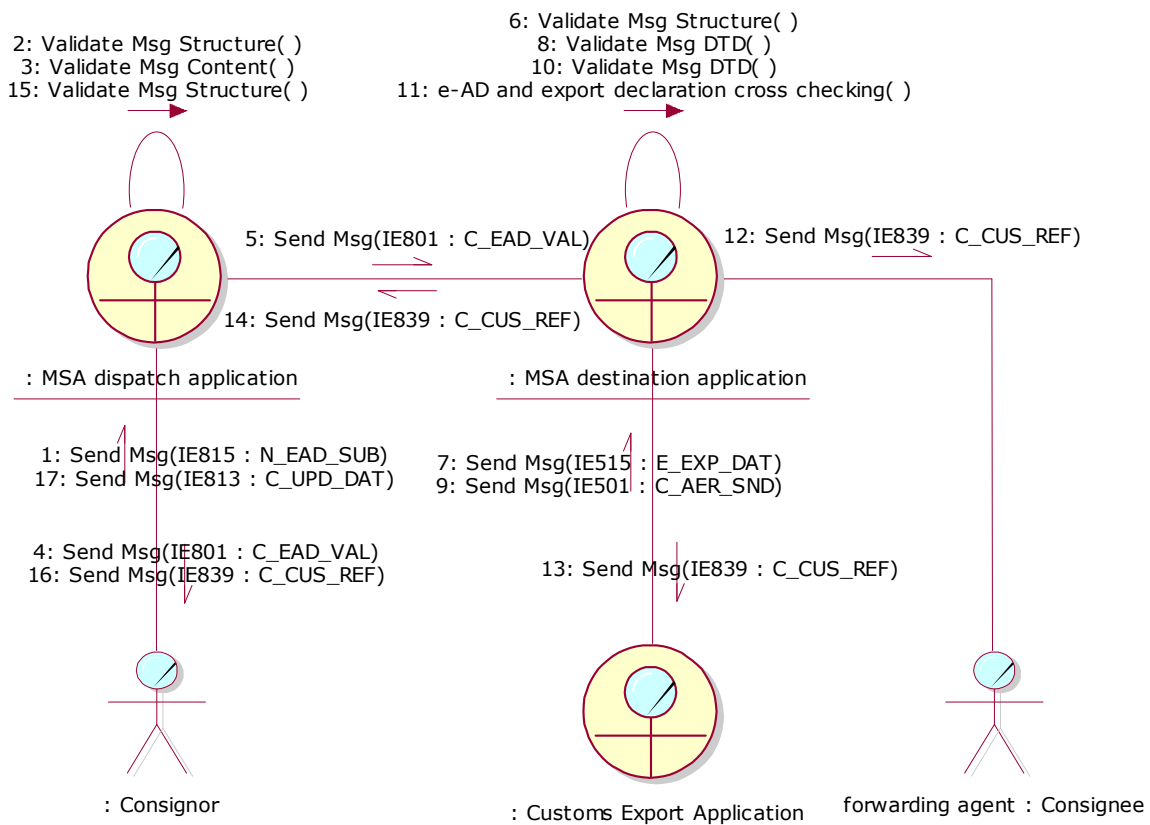


Figure 103: CLD - Export Operation at Office of Export followed by cross-checking failure after the export release, export declaration cancellation and submission of change of destination

III.II.3.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration

In this scenario, the movement has not been released by Customs, and therefore the concerned e-ADs have been maintained in the “Accepted” state. Following, the forwarding agent submits a new export declaration.

This scenario may be followed by:

- Positive cross-checking, release by Customs and export confirmation of exit (Section III.II.2.1 Export Operation at Office of Export followed by Export confirmation of exit);
- Positive cross-checking, release by Customs, and export cancellation of exit (Section III.II.2.2 Export Operation at Office of Export followed by Export Cancellation of exit);

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- Negative cross-checking, export declaration cancellation and submission of new export declaration (Section III.II.2.4 Export Operation at Office of Export followed by negative cross-checking *before* the export release and Change of Destination, Section III.II.2.5 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of new export declaration, Section III.II.2.6 Export Operation at Office of Export followed by negative cross-checking *after* the export release, export declaration cancellation and submission of change of destination);
- No release by Customs (Section III.II.2.7 Export Operation at Office of Export and movement not released by Customs followed by new export declaration, Section III.II.2.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination).

This scenario is depicted in Figure 104 and Figure 105:

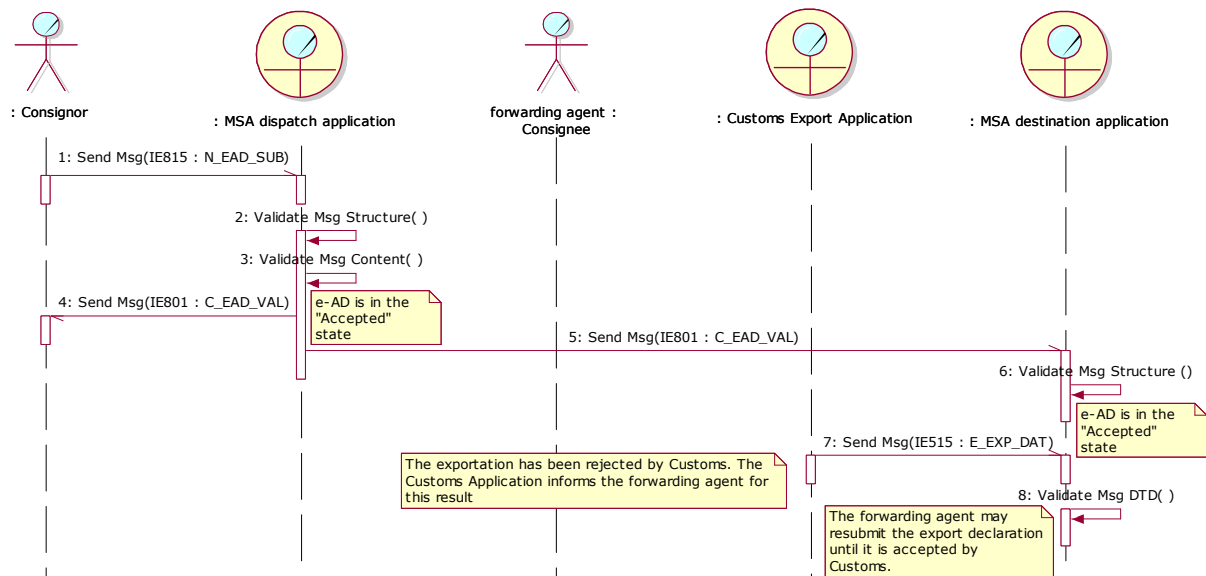


Figure 104: TSD - Export Operation at Office of Export and movement not released by Customs followed by new export declaration

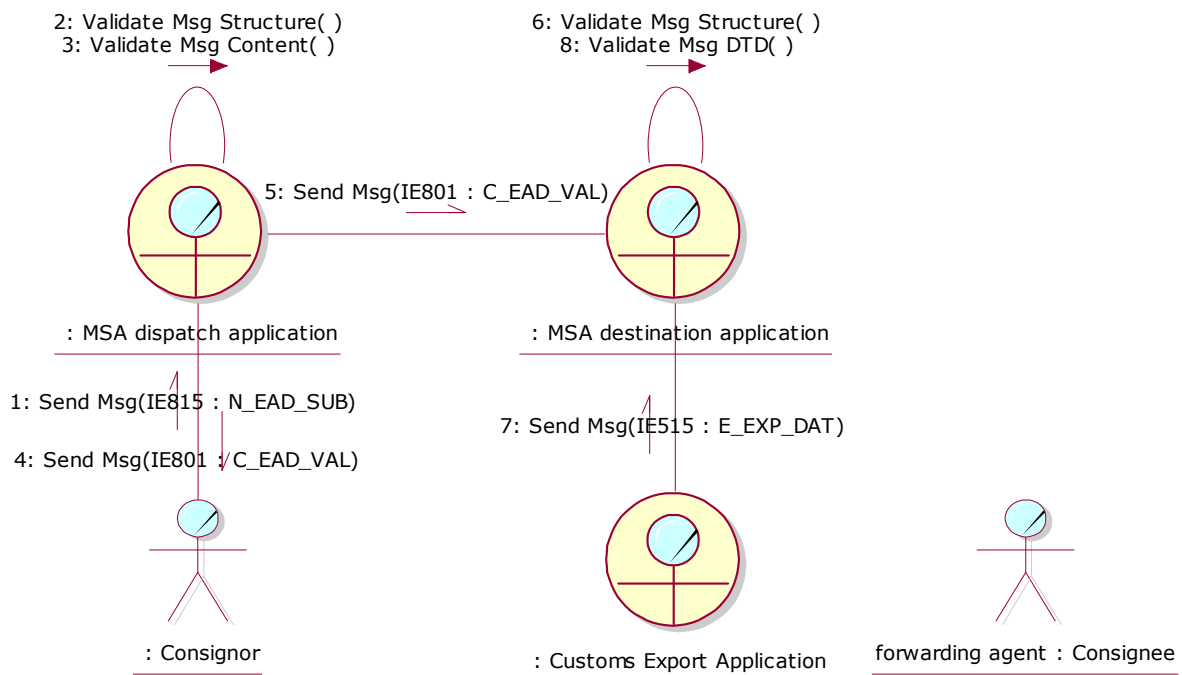


Figure 105: CLD - Export Operation at Office of Export and movement not released by Customs followed by new export declaration

III.II.3.8 Export Operation at Office of Export and movement not released by Customs followed by change of destination

In this scenario, the movement has not been released by Customs, and therefore the e-ADs have been maintained in the “Accepted” state. Following, the Consignor issues a change of destination.

This scenario is depicted in Figure 106 and Figure 107:

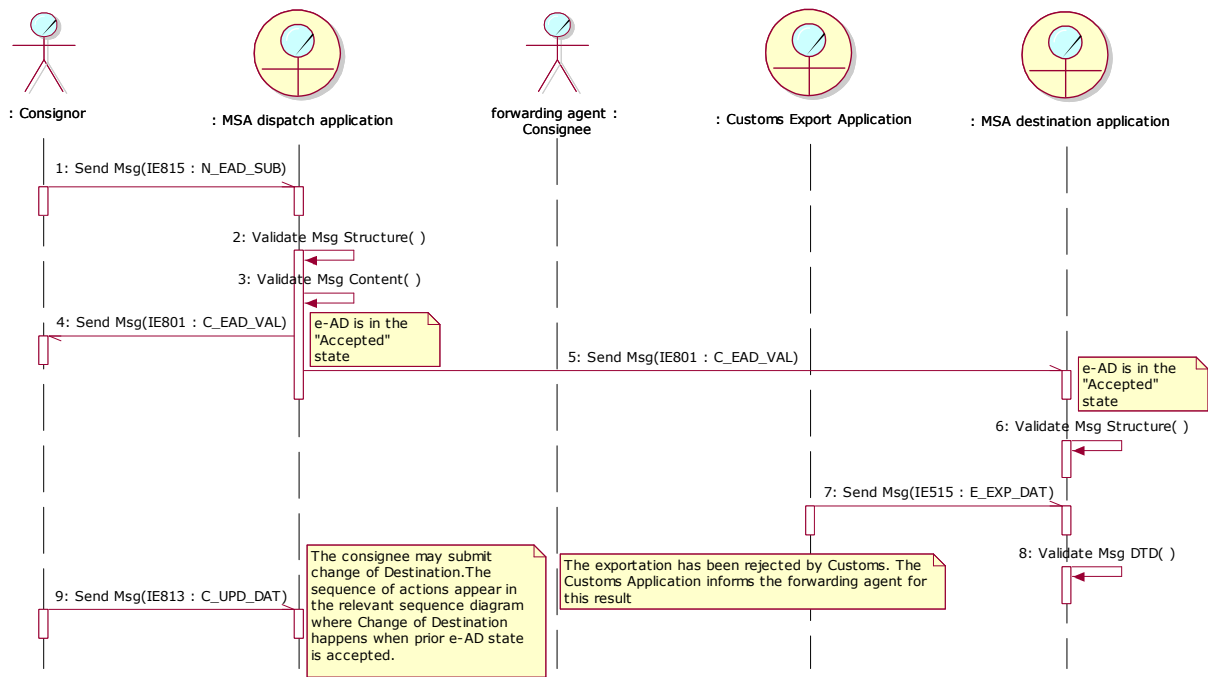


Figure 106: TSD - Export Operation at Office of Export and movement not released by Customs followed by change of destination

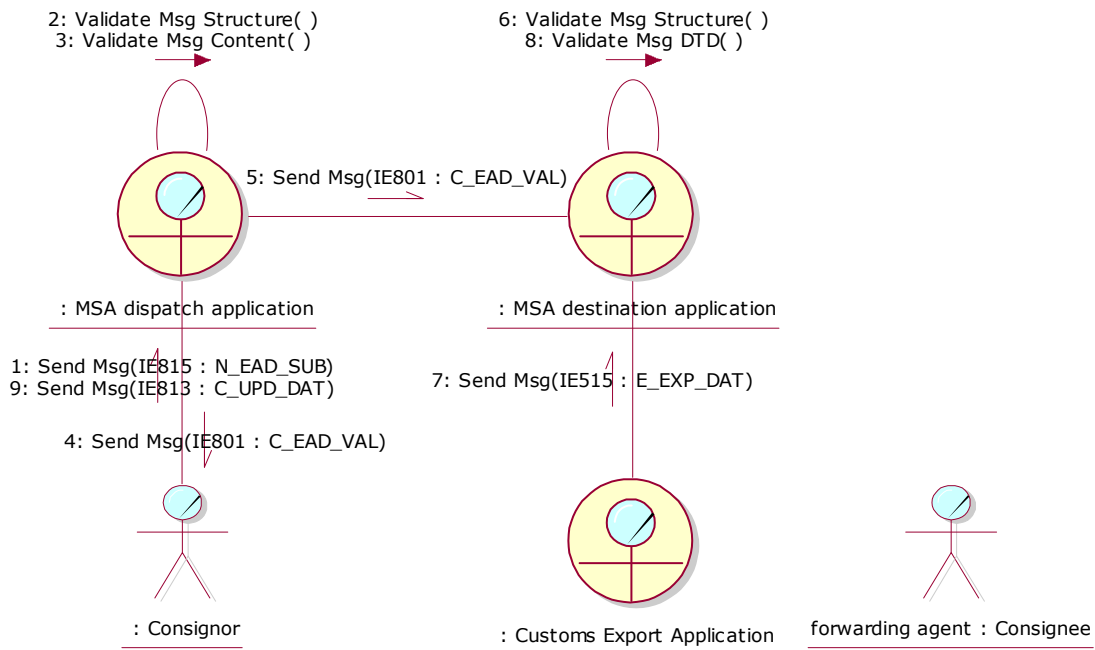


Figure 107: CLD - Export Operation at Office of Export and movement not released by Customs followed by change of destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Sub-Section III.III State-Transition Diagrams for Central Circuit Scenarios

III.III.1 STD at Dispatch

The following State Transition Diagrams in Figure 108: , Figure 109: and Figure 110: depict the State Machine at the MSA of Dispatch. All the transitions are described in detail in Chapter Central Circuit Scenarios. In addition, the state transition to the “*stopped*” state is described in section IV.I.3 Interruption of a movement (UC3.05).

The STDs embody conditions for the value of the Sequence Number in the incoming messages. The condition for the Sequence Number is indicated in Square Brackets next to the incoming message. The notion SeqNo and PreviousSeqNo is used to represent the Sequence Number in the incoming message and the last Sequence Number known at the receiving MSA. It is noted that the IE807, IE813, IE815 and IE825 messages do not include the Sequence Number and therefore there is no corresponding condition for their Sequence Number.

For the case of the IE810, the STD in Figure 108: embodies conditions for the value of the Sequence Number (PreviousSeqNo), the existence of an Upstream ARC, the existence and the value of the Deferred submission flag (DefSubmFlag), indicated also in Square Brackets next to the message. Since the IE810 does not include a Sequence Number, the condition for the Sequence Number is validated against the last Sequence Number known at the receiving MSA. Respectively, the condition for the existence of the Upstream ARC is validated against the latest IE801 received at the receiving MSA, given that the IE810 does not include an Upstream ARC. Finally, the condition for the existence and the value of the Deferred submission flag is validated against the draft e-AD, since it consists a field present only in this message.

It should be noted that the following STD and each STD concerns one e-AD. For each new change of destination (Section III.I.5) the IE813 message will lead to the increase of the Sequence Number of the ARC, but the ARC will remain the same. Hence, the same STD will host the state transitions of a specific ARC.

In the case of splitting (Section III.I.6) the original ARC undergoes a state transition to “Replaced” state, which is a final state and the new ARCs will be placed in separate STDs.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

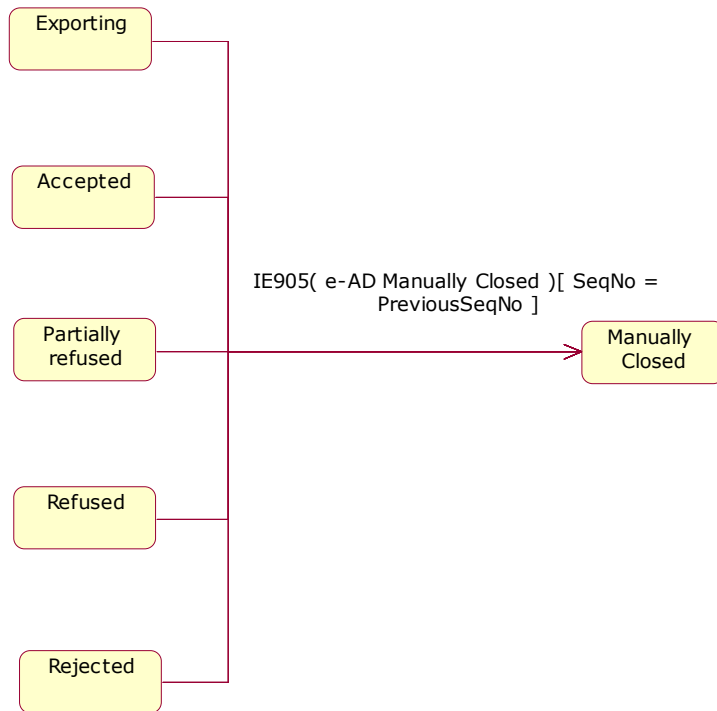


Figure 110: STD at MSA of Dispatch for Central Circuit Scenarios – part 3

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.III.2 STD at Destination

The following State Transition Diagrams in Figure 111: , Figure 112: and Figure 113: depict the State Machine at the MSA of Destination. All the transitions are described in detail in Chapter Central Circuit Scenarios. In addition, the state transition to the “*stopped*” state is described in section IV.I.3 Interruption of a movement (UC3.05).

The STDs embody conditions for the value of the Sequence Number in the incoming messages. The condition for the Sequence Number is indicated in Square Brackets next to the incoming message. The notion SeqNo and PreviousSeqNo is used to represent the Sequence Number in the incoming message and the last Sequence Number known at the receiving MSA. It is noted that the IE807 message does not include the Sequence Number and therefore there is no corresponding condition for its Sequence Number. In addition, the Sequence Number in an incoming IE801 with a new ARC (transition from the standard start state to the “Accepted” state) can be anything and therefore the Sequence Number is not subject to a condition.

For the case of the IE810, the STD in Figure 111: embodies conditions for the value of the Sequence Number (PreviousSeqNo) and the existence of an Upstream ARC, indicated also in Square Brackets next to the message. Since the IE810 does not include a Sequence Number, the condition for the Sequence Number is validated against the last Sequence Number known at the receiving MSA. Respectively, the condition for the existence of the Upstream ARC is validated against the latest IE801 received at the receiving MSA, given that the IE810 does not include an Upstream ARC.

It should be noted that the following STD and each STD concerns one e-AD. Hence, the same STD will host the state transitions of a specific ARC. For each new change of destination (Section III.I.5) the IE813 message will lead to the increase of the Sequence Number of the ARC, but the ARC will remain the same.

The state transitions from “Rejected” or “Refused” or “Partially refused” to “Accepted” occurs with the increase of the previous Sequence Number by one.

The state transition from “Diverted” to “Accepted” occurs with the increase of the previous Sequence Number at least by two.

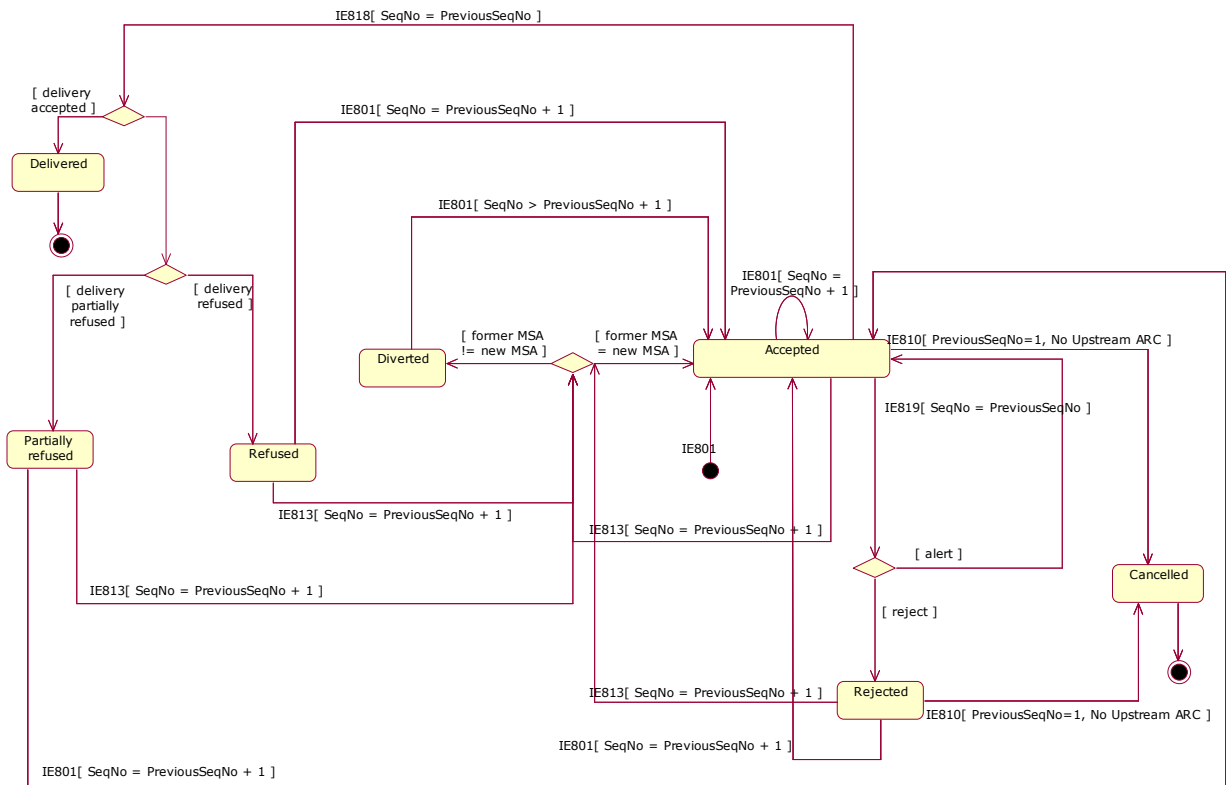


Figure 111: STD at MSA of Destination for Central Circuit Scenarios – part 1

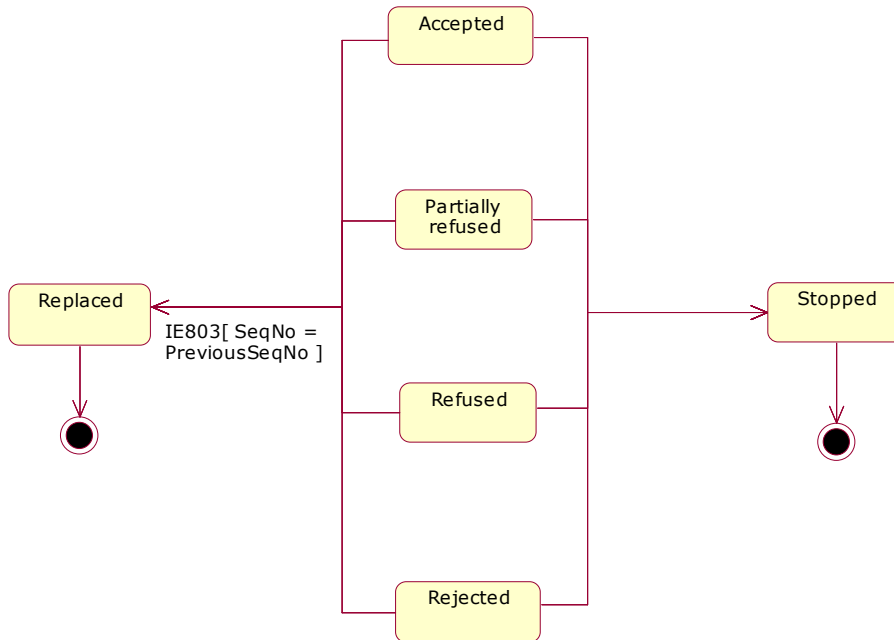


Figure 112: STD at MSA of Destination for Central Circuit Scenarios – part 2

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

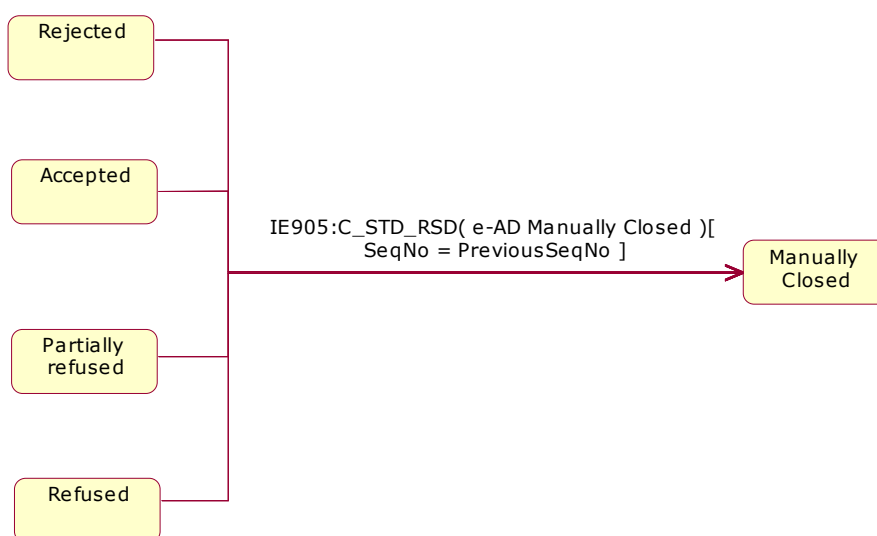


Figure 113: STD at MSA of Destination for Central Circuit Scenarios – part 3

Sub-Section III.IV State-Transition Diagrams for Export Scenarios

The following State Transition Diagrams depict the State Machine at the MSA of Dispatch. All the transitions are described in detail in Sub-Section III.II. There is a need to distinguish between the two export Use Cases since the state transitions are different.

Although the State Transition Diagrams of this section show the IE501 (E_AER_SND) and the IE518 (C_EXT_RES) messages, it should be noted that these messages do not cause the actual state transitions within EMCS. It is the NDEAs that perform the state transitions after successful completion of the relevant operations triggered by the reception of an IE501 (E_AER_SND) or an IE518 (C_EXT_RES) messages according to the scenarios of Sub-Section III.II.

The STDs embody conditions for the value of the Sequence Number in the incoming messages. The condition for the Sequence Number is indicated in Square Brackets next to the incoming message. The notion SeqNo and PreviousSeqNo is used to represent the Sequence Number in the incoming message and the last Sequence Number known at the receiving MSA.

III.IV.1 Local Clearance at Export

III.IV.1.1 STD at Dispatch

It is noted that the IE807, IE810, IE813, IE815, IE825, IE518 and IE501 messages do not include the Sequence Number and therefore there is no corresponding condition for their Sequence Number.

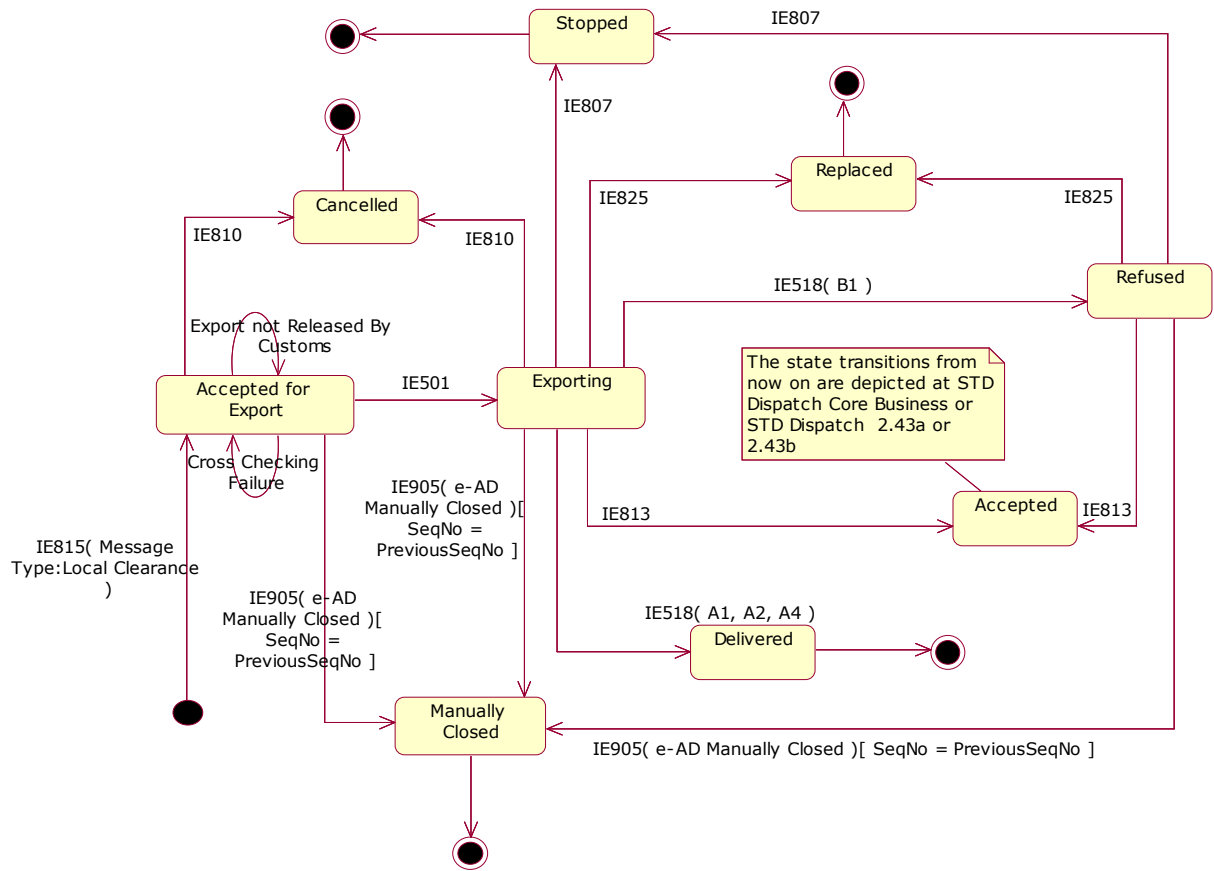


Figure 114: STD at Dispatch - Local Clearance

III.IV.1.2 STD at Destination

The destination side does not exist in this case so a State Transition Diagram is not applicable.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.IV.2 Export operation at Office of Export

III.IV.2.1 STD at Dispatch

III.IV.2.1.1 When MS of Dispatch is MS of Export as well

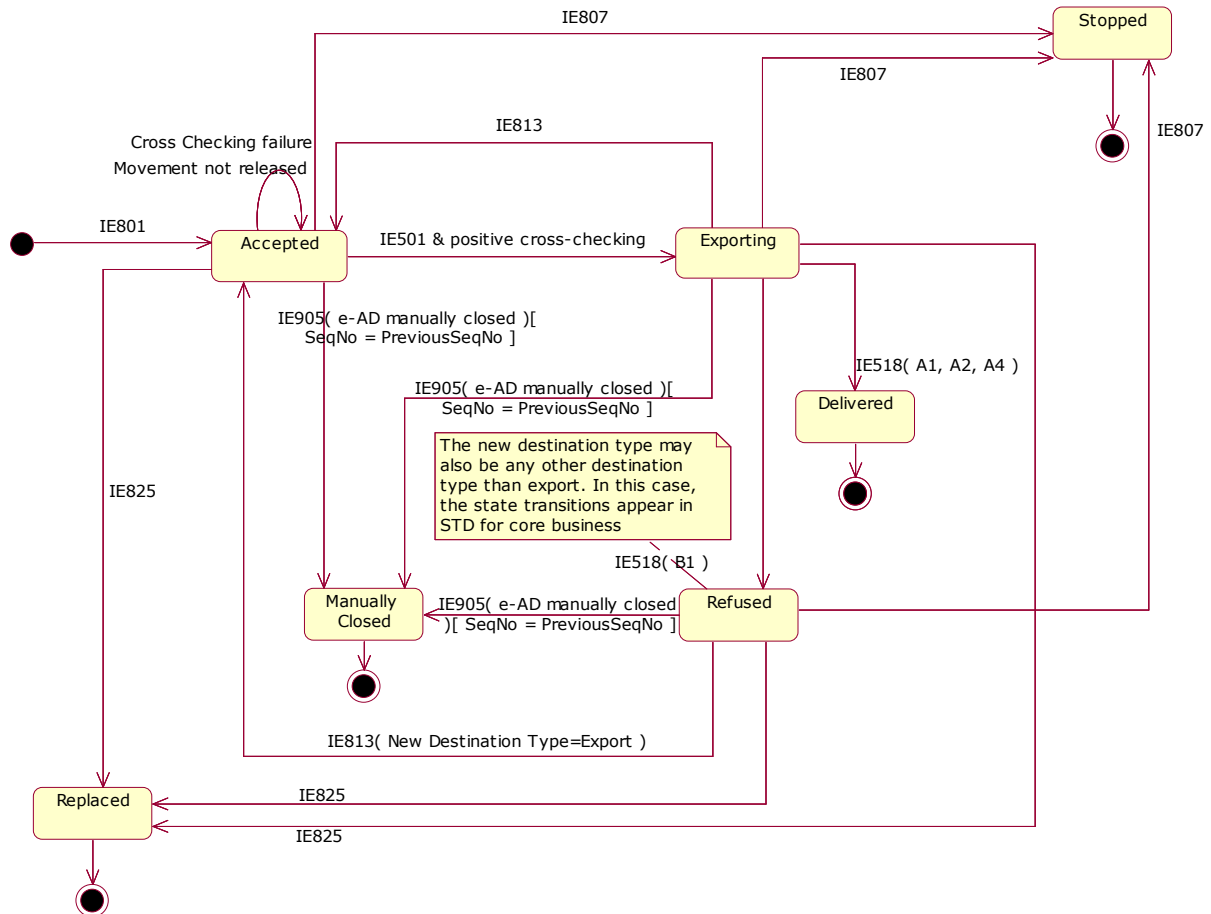


Figure 115: STD at Dispatch - Export operation at Office of Export when MS of Dispatch same as MS of Export

III.IV.2.1.2 When MS of Dispatch is different than MS of Export

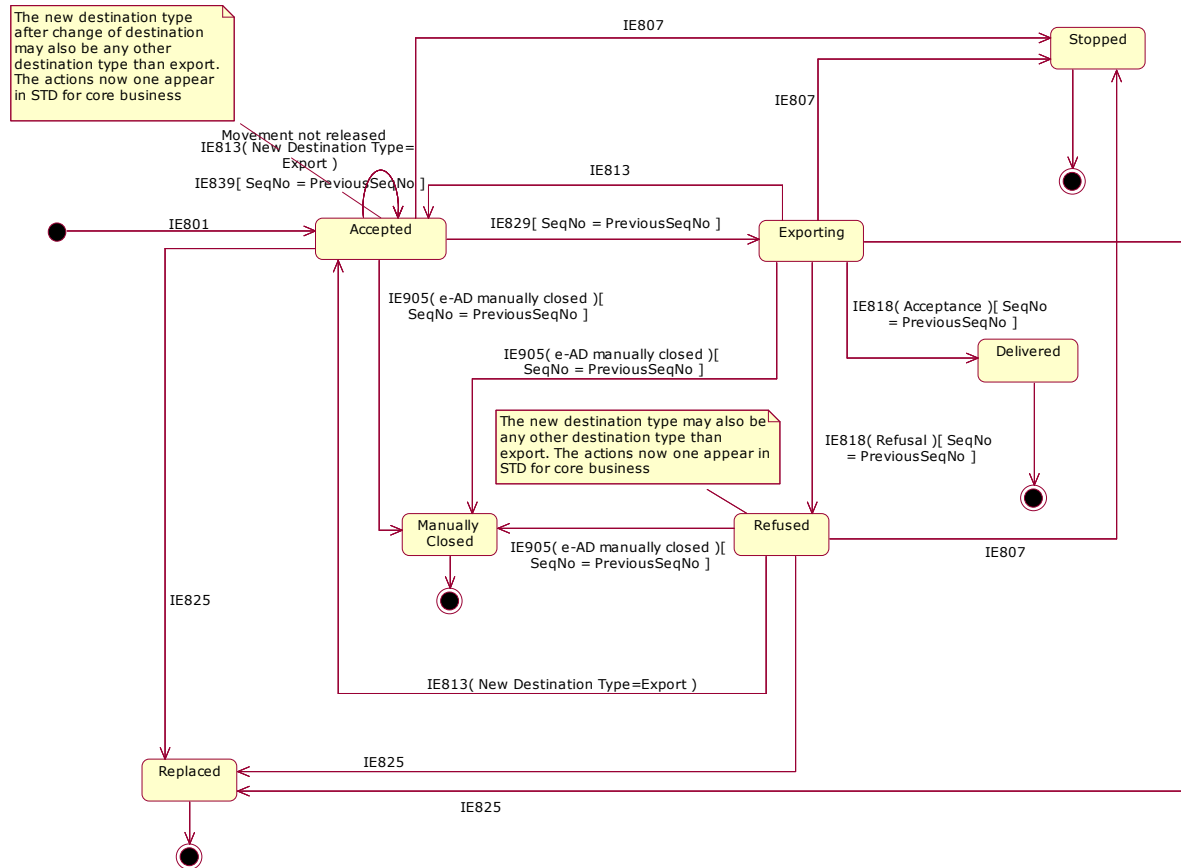


Figure 116: STD at Dispatch - Export operation at Office of Export when MS of Dispatch is different than MS of Export

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.IV.2.2 STD at Destination

III.IV.2.2.1 When MS of Dispatch is MS of export as well

There is no destination application when MS of dispatch is MS of export as well.

III.IV.2.2.2 When MS of Dispatch is different than MS of Export

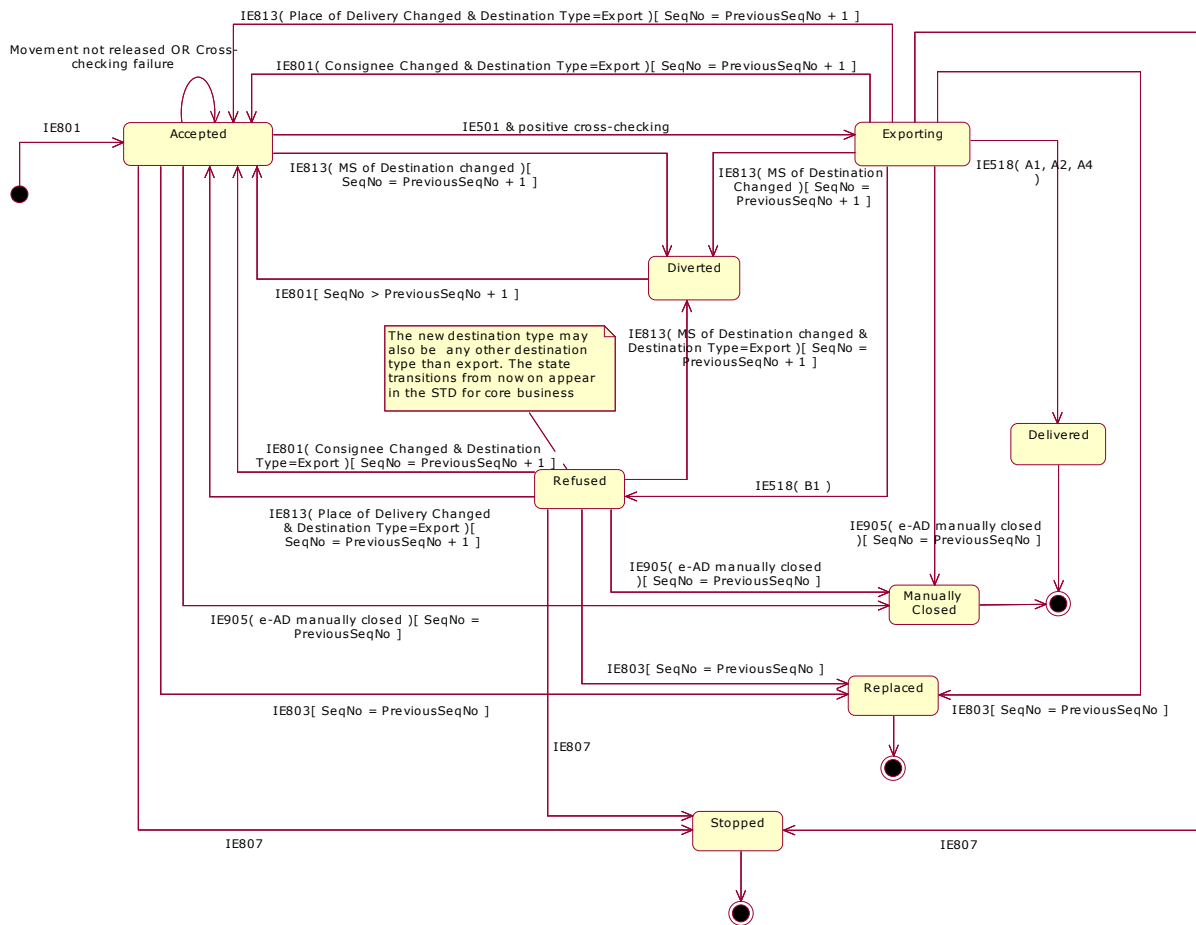


Figure 117: STD at Destination - Export operation at Office of Export when MS of Dispatch is different than MS of Export

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Sub-Section III.V Functional Timers

TIM_EAD	
Started:	<p>UC-201-230 - Start Follow up When the submitted e-AD is valid, the MSA dispatch application starts the TIM_EAD timer to expire at the expected end of the movement, which is the date of dispatch plus the journey time.</p>
Stopped:	<p>UC-206-410 - Receive Report of Receipt at MSA of dispatch When the MSA dispatch application receives a Report of Receipt for an accepted delivery, then it stops the timer TIM_EAD.</p> <p>UC-210-220 - Validate cancellation of e-AD When the cancellation of e-AD is accepted by MSA dispatch application and the TIM_EAD still runs, then the MSA dispatch application stops the timer.</p>
Reset:	<p>UC-206-410 - Receive Report of Receipt at MSA of dispatch When the MSA dispatch application receives a Report of Receipt for an accepted delivery and the TIM_EAD has already expired at the limit date, then the MSA dispatch application resets the flag.</p> <p>UC-205-230 - Start Follow up When the journey time of an e-AD is updated after a change of destination and the TIM_EAD timer has expired, the MSA dispatch application resets the flag that has been raised locally at expiration time. In addition, the MSA dispatch application restarts the timer (TIM_EAD) by setting as new expected end date the updated journey time if and only if the new expected end date is later than the present date.</p> <p>UC-210-220 - Validate cancellation of e-AD When the cancellation of e-AD is accepted by MSA dispatch application and the TIM_EAD has already expired at the limit date, then the MSA dispatch application resets the flag.</p>
Update:	<p>UC-205-230 - Start Follow up When the journey time of an e-AD is updated after a change of destination and considering that TIM_EAD has not expired, the MSA dispatch application updates the timer with the new expected end date.</p>

Table 15: TIM_EAD functional timer

TIM_FDF	
Started:	<p>UC-201-230 - Start Follow up In case an e-AD has been accepted and the destination fields have been left empty according to Article 22 of Directive 2008/118/EC, the TIM_FDF timer is initiated by the MSA dispatch application to expire at the limit date for filling in the destination fields.</p>

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Stopped:	UC-205-230 - Start follow up After the change of destination and if the concerned e-AD has been accepted without destination details (Article 22 of Directive 2008/118/EC), the MSA dispatch application stops the TIM_FDF timer, since the destination has been specified.
Reset:	UC-205-230 - Start follow up In case the timer TIM_FDF has expired before the change of destination for the e-AD with empty destination fields, the MSA dispatch application resets the flag.

Table 16: TIM_FDF functional timer

TIM_CHS	
Started:	UC-206-410 - Receive Report of Receipt at MSA of dispatch When the MSA dispatch application receives a Report of Receipt for a refused delivery, then it initiates the TIM_CHS timer to expire at limit date for submission of a change of destination.
Stopped:	UC-205-230 - Start follow up The MSA dispatch application stops the timer when the destination for a movement of which delivery is refused, has changed.
Reset:	UC-205-230 - Start follow up When the timer has already expired and the destination changes for a movement of which delivery is refused, the MSA dispatch application resets the flag for this movement.

Table 17: TIM_CHS functional timer

Sub-Section III.VI Exception Handling

This section describes the mechanisms that shall be available for handling exceptional cases in the Common Domain.

III.VI.1 Rejection due to functional errors

This section describes a mechanism for handling exceptional cases in the Common Domain occurring due to functional errors. These functional errors are described in detail in Chapter VIII.I.3.2.3 Semantic layer of Section VIII. The scenarios in Chapter VIII.I.3.2.3 Semantic layer of Section VIII are indicative and aim to provide some information regarding the usage of the following functional errors:

- An element may contain a value, which is not part of the applicable Codelist;
- An element may contain a value, which has exceeded the maximum or minimum length specified;
- The message contains an ARC which has an invalid structure;
- A Data Group may have too many repetitions than allowed;
- A mandatory/required element is missing from the message;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The received message is out of sequence;
- The received message contains an ARC, which is unknown to the MSA (no e-AD with the specific ARC exists);⁵
- The received message has the same Message identification (MsgID) with the one received before Duplication is detected.

III.VI.2 Technical Mechanism of IE904/IE905

This Section describes the message exchange protocol for the IE904/IE905 mechanism. The mechanism is used as follows:

- manual status request/response: manually triggered to request only the status of a particular e-AD; or
- manual status synchronisation request: manually triggered to request the status of a particular e-AD as well as the last message sent; or
- automatic status synchronisation request: same as manual status synchronisation request, but automatically triggered in order to identify whether an e-AD/RoR is missing in particular scenarios.

This Section describes all of the aforementioned cases by providing generic scenarios (in Section III.VI.2.1) describing the message exchange protocol for the manually triggered IE904/IE905 mechanism. By generic scenarios it is meant that MSA applications involved in the IE904/IE905 message exchange should not assume any particular roles or special assumptions except from whether the e-AD is known or unknown. Therefore, in Section III.VI.2.1 the roles used is the general role “MSA application” instead of specific roles such as MSA of Dispatch or the MSA of Destination.

Furthermore, this Section provides indicative scenarios (in Sections III.VI.2.2 and III.VI.2.3) for the manually triggered IE904/IE905 mechanism by applying the mechanism in the context of particular scenarios. Finally, this Section describes the message exchange protocol for the automatically triggered IE904/IE905 mechanism (in Section III.VI.2.4). The aforementioned Sections (III.VI.2.2, III.VI.2.3, and III.VI.2.4) use specific roles such as the MSA of Dispatch or the MSA of Destination.

It should be noted that the IE904/IE905 mechanism complement to the aforementioned cases, it is also used in the Download of an e-AD (UC2.51) Use Case (see III.I.12).

⁵ *This functional error shall not be used from the MSA destination application for the case where a draft Report of Receipt is submitted by the Consignee containing an ARC unknown to MSA of Destination. This case is handled using the IE904/IE905 mechanism.*

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.1 Generic scenarios

This Section describes the general message exchange protocol for the IE904/IE905 mechanism independent of any preconditions other than the ARC being known or unknown. The roles that are used in the following scenarios do not concern specific roles such as the MSA of Dispatch or the MSA of Destination. Rather, the general role “MSA application” is used. This decouples the roles that the applications may have for a given ARC queried by the IE904/IE905 mechanism.

III.VI.2.1.1 Manual Status Request/Response

This Section describes the Status Request/Response mechanism that *could* be used at any time by any MSA application to request the status of an e-AD from any MSA application, independent of any preconditions other than the ARC being known or unknown. Several information exchanges of an excise operation (identified by a unique ARC) require a timely response. Due to exceptions, those responses may not arrive in time.

The Status Request (IE904: C_STD_REQ) is triggered manually by the MSA Official for a specific ARC existing in the system. The MSA Official shall indicate that he/she is simply enquiring about the status of the movement. In the IE904: C_STD_REQ the MSA Official of the Requesting MSA application shall also provide the sequence number of the last business (event) message sent to the Requested MSA application for the specific ARC.

More specifically the IE904 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC provided by the Excise Officer of the Requesting MSA application;
- The “Sequence Number”: It includes:
 - the value “1” if the specific ARC is unknown to the Requesting MSA application;
OR
 - the last known sequence number for the specific ARC;
- The “Status”: It includes:
 - “None” if the specific ARC is unknown to the Requesting MSA application; *OR*
 - the State for the specific movement in the Requesting MSA application;
- The “Last Received Message Type”: It includes:
 - “None” if the specific ARC is unknown to the Requesting MSA application or the Requesting MSA application has *not* received a business (event) message from the Requested MSA; *OR*
 - the number of the last business (event) message sent from the Requested MSA application and received by the Requesting MSA application for the specific ARC;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The “Status Request Message Type”: This item is not applicable in this case that the IE904 message is sent for a Status Request/Response scenario.

The following sections describe the scenarios and the response of the Requested MSA application depending on whether the ARC included in the IE904 sent by the Requesting MSA application is known (III.VI.2.1.1.1) or unknown (III.VI.2.1.1.2) to the Requested MSA application.

III.VI.2.1.1.1 Manual Status Request/Response for a known ARC

The Requested MSA application examines the IE904: C_STD_REQ message content to identify what the Requesting MSA application is expecting in reply. In this scenario the IE904: C_STD_REQ reports a simple status request (the “Status Request Message Type” Data Item is not present) and the Requested MSA application replies with a single message, the Status Response (IE905: C_STD_RSP).

In particular, the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC of the specific movement in the Requested MSA application. It is the same as the ARC of the request (IE904) message;
- The “Sequence Number”: It includes the latest Sequence Number for the specific ARC in the Requested MSA application;
- The “Status”: It includes the State for the specific ARC in the Requested MSA application;
- The “Last Received Message Type”: It includes:
 - “None” if the specific ARC is unknown to the Requested MSA application or the Requested MSA application has *not* received a business (event) message from the Requesting MSA; *OR*
 - the number of the last business (event) message sent from the Requesting MSA application and received by the Requested MSA application for the specific ARC;

The scenario of Manual Status Request/Response for a known ARC is depicted in Figure 118 and Figure 119.

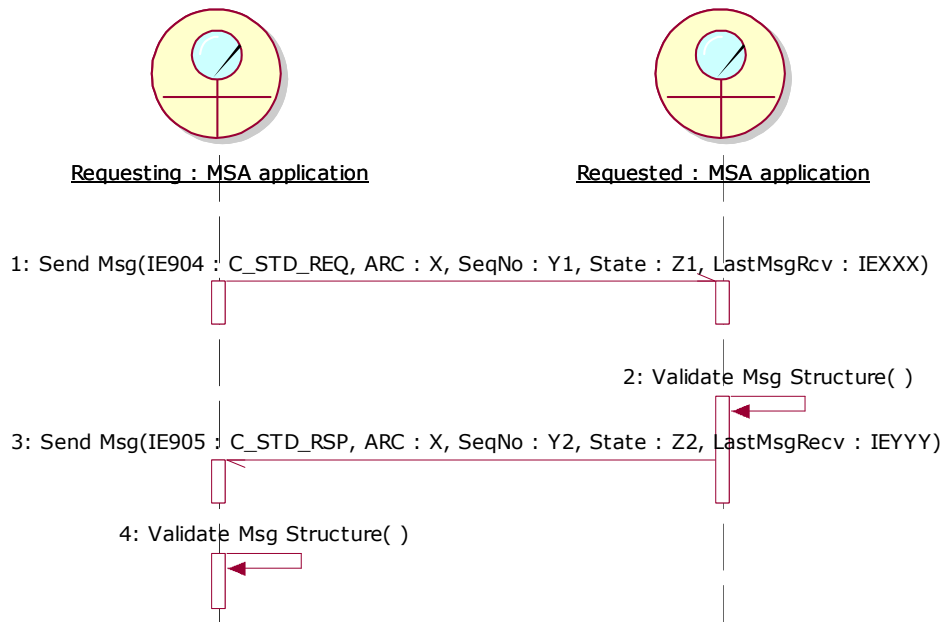


Figure 118: TSD - Manual Status Request/Response for a known ARC

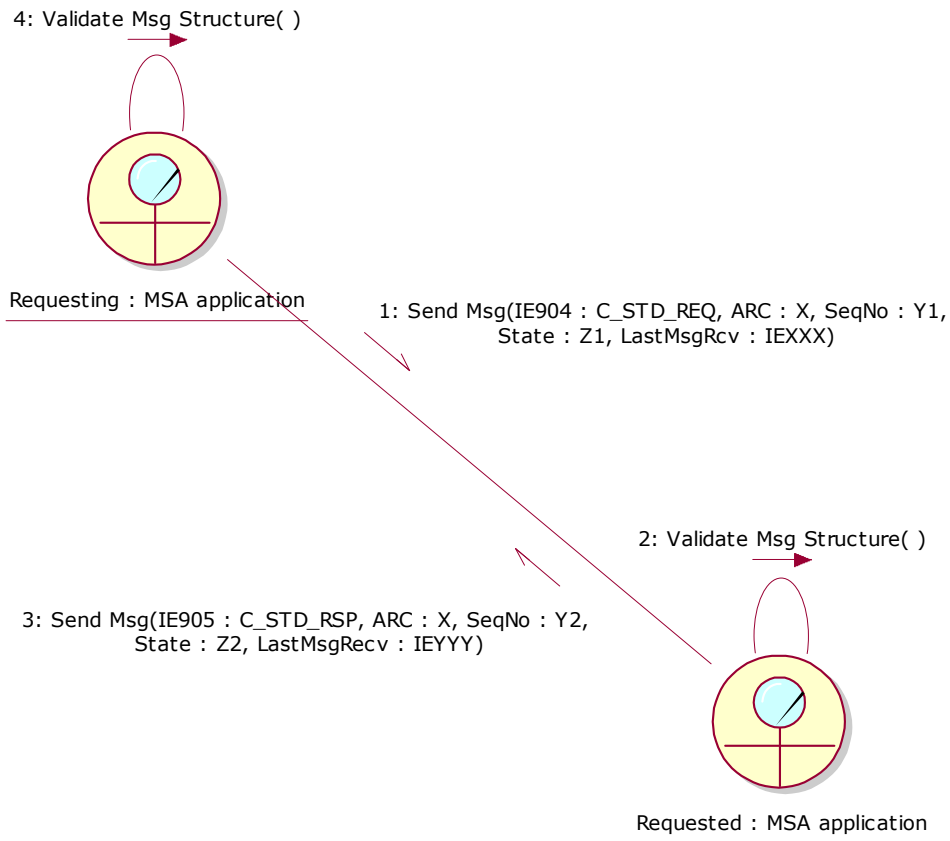


Figure 119: CLD - Manual Status Request/Response for a known ARC

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.1.1.2 Manual Status Request/Response for an unknown ARC

The Requested MSA application examines the IE904: C_STD_REQ message content to identify what the Requesting MSA application is expecting in reply. In this scenario the IE904: C_STD_REQ reports a simple status request (the “Status Request Message Type” Data Item is not present) and the Requested MSA application replies with a single message, the Status Response (IE905: C_STD_RSP).

In this scenario, the ARC included in the IE904: C_STD_REQ is unknown to the Requested MSA application, so the Status Response (IE905) should contain “None” as status and “None” as the last business message received from the Requesting MSA application.

Therefore, when an MSA receives a Status Request/Response (IE904) for an unknown e-AD, it should not refuse it with <FUNCTIONAL ERROR.Error Type> equal to error code “Unknown ARC” via an IE906 message.

In particular, the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC provided in the request (IE904) message;
- The “Sequence Number”: It includes the value “1”, since the specific ARC is unknown to the Requested MSA application;
- The “Status”: It includes the value “None”, since the specific ARC is unknown to the Requested MSA application.
- The “Last Received Message Type”: It includes the value “None”, since the specific ARC is unknown to the Requested MSA application.

The scenario of Manual Status Request/Response for an unknown ARC is depicted in Figure 120 and Figure 121.

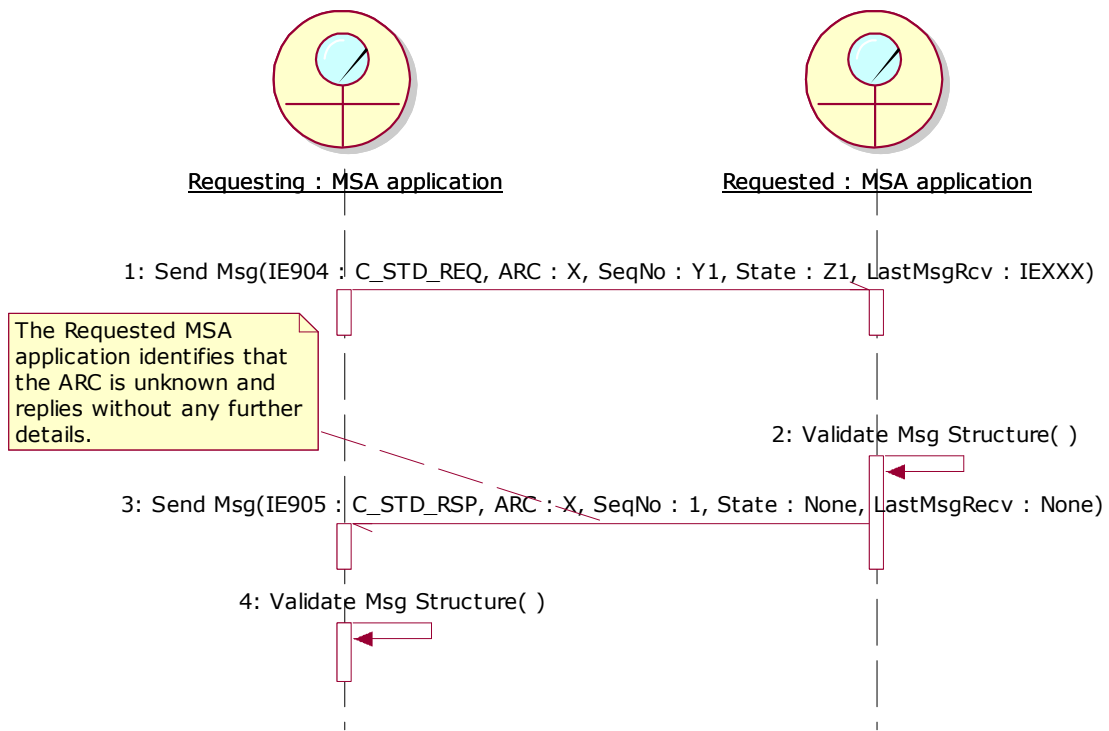


Figure 120: TSD - Manual Status Request/Response for an unknown ARC

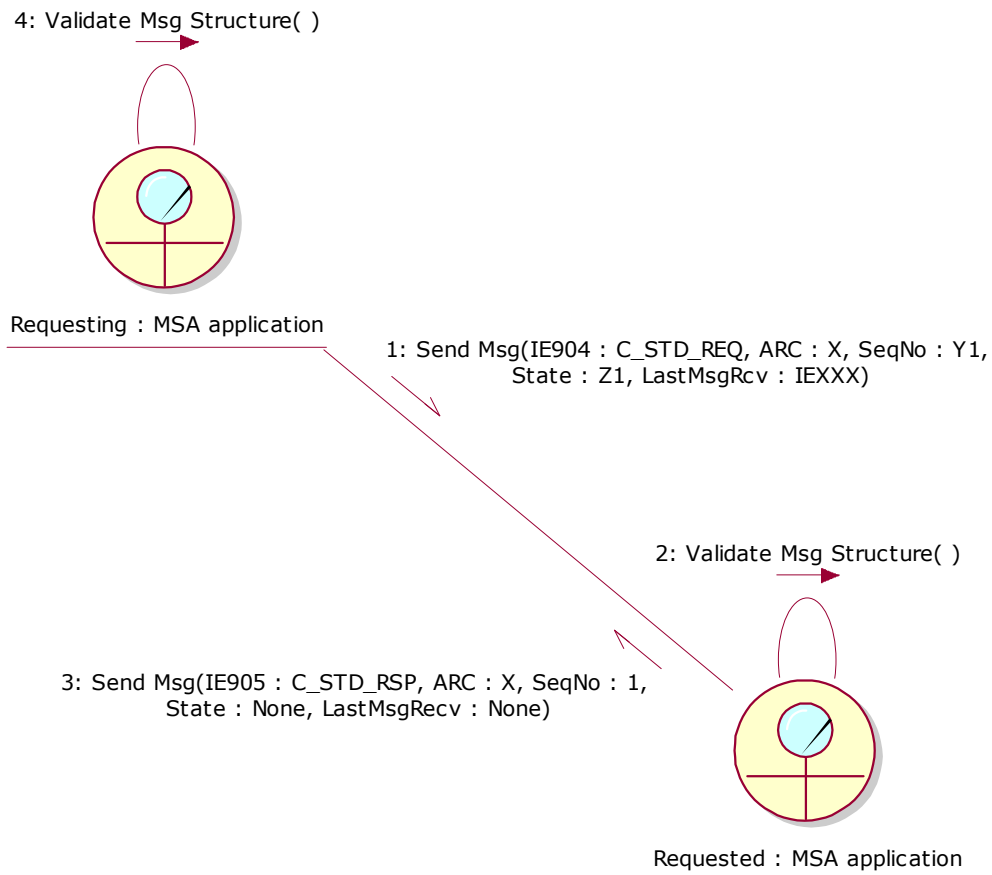


Figure 121: CLD - Manual Status Request/Response for an unknown ARC

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.1.2 Manual Status Synchronisation Request

In Section III.VI.2.1.1 Manual Status Request/Response, the MSA Official issues a simple status request, reporting that he/she wishes to receive in reply only the movement status at the communicating MSA. This Section presents the “Manual Status Synchronisation Request” mechanism, which *could* be provided by a MSA application to its MSA Officials as a more advanced (sophisticated) use of the “Manual Status Request/Response”. In particular, the “Manual Status Synchronisation Request” is used to indicate that the MSA Official requests not only the status of the movement but also the last message sent from the Requested MSA application.

More specifically the IE904 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC provided by the Excise Officer of the Requesting MSA application;
- The “Sequence Number”: It includes:
 - the value “I” if the specific ARC is unknown to the Requesting MSA application; *OR*
 - the last known sequence number for the specific ARC;
- The “Status”: It includes:
 - “None” if the specific ARC is unknown to the Requesting MSA application; *OR*
 - the State for the specific movement in the Requesting MSA application;
- The “Last Received Message Type”: It includes:
 - “None” if the specific ARC is unknown to the Requesting MSA application or the Requesting MSA application has *not* received a business (event) message from the Requested MSA; *OR*
 - the number of the last business (event) message sent from the Requested MSA application and received by the Requesting MSA application for the specific ARC;
- The “Status Request Message Type”: This item will have the value “I” indicating that the Excise Officer of the Requesting MSA application is triggering a Status Synchronisation Request.

The following sections describe the scenarios and the response of the Requested MSA application depending on whether the ARC included in the IE904 sent by the Requesting MSA application is known (III.VI.2.1.2.1) or unknown (III.VI.2.1.2.2) to the Requested MSA application

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

It should be noted that for both of the aforementioned scenarios, the resubmission of a message is referring to the regeneration of the “Common Message Header” of the message. More specifically, the lost/delayed messages are submitted again, hence with the same business content (same “Body” as defined in Section VII.I.4 “Common message header”), but with different content in the “Common Message Header” of the message (including a new unique “Message Identifier” Data Item). More specifically, the “Common Message Header” Data Group should conform to the guidelines of Section VII.I.4 and consist of the following values in the relevant Data Items:

- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items should conform to the guidelines of Section VII.I.4. The values of the aforementioned Data Items of the regenerated message (sent after the response IE905 message due to the Status Synchronisation Requests, in order to replace the lost/delayed messages) should have the same values as the original message;
- The “Message identifier”: This is a “Required” Data Item generated by the sending application to uniquely identify the information exchange. The “Message Identifier” value of the message sent due to Status Synchronisation Request should also be unique. Therefore, it will differ from the value of the original message (lost/delayed message);
- The “Correlation identifier”: In order to correlate the lost/delayed messages to the Status Synchronisation Request, the “Correlation identifier” Data Item should equal the value of the “Message identifier” Data Item of the IE904 message. Therefore, it will differ from the value of the original message (lost/delayed message).

When a Status Synchronisation Request is performed, the lost/delayed message is retransmitted. The Requesting MSA application triggering the Status Synchronisation Request might either be synchronised with the Requested MSA application or not:

- If the Requesting and the Requested MSA applications are synchronised, it leads to the conclusion that the lost/delayed message has already been received by the Requested MSA application. In this case, the Requesting MSA application after receiving the (alleged) lost/delayed message should ignore it, since it has already been processed;
- If the Requesting and the Requested MSA applications are not synchronised, then the Requesting MSA application after receiving the lost/delayed message should process it in order to become synchronised with the Requested MSA application.

In order to achieve technically this correlation, the IE905 and the lost/delayed message should be correlated to the IE904, thus enabling the Requesting MSA application to handle the lost/delayed message differently. The correlation of the IE905 and the lost/delayed message is performed by the “Correlation Identifier” Data Item, which will be equal to the “Message Identifier” of the IE904 not only for the IE905, but also for the lost/delayed message. So, whenever a business message has a value in the “Correlation Identifier” Data Item, this signifies that the message is sent as a result of a Status Synchronisation Request and that it should be treated differently. For such messages there are two possible cases:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The message has already been received, so it can be safely ignored (in this case the two applications involved are already synchronised prior to the Status Synchronisation Request);
- The message has not been received, so it can be processed normally (in this case the two applications involved are not synchronised, but they become synchronised after the Status Synchronisation Request).

In the exceptional case where the reception of the original message has been delayed and it is received after a Status Synchronisation Request, then the original message will unavoidably be refused with a “Message out of sequence” error (IE906 message with code “92” in the “FUNCTIONAL ERROR.Error Type” Data Item). However, the requested application will not be alarmed since it is aware that the message has been sent one time normally, and a second time due to Status Synchronisation Request. Hence, it realises that although the original message has been refused, the Requesting MSA application has processed the message received as a result of the Status Synchronisation Request.

For the cases of duplicate messages due to application fault, the state machine should be followed. When a message is unexpected according to state machine, the MSA applications should reject them with an IE906 message indicating a “Message out of sequence” error.

III.VI.2.1.2.1 Manual Status Synchronisation Request for a known ARC

The Requested MSA application examines the IE904: C_STD_REQ message content to identify what the Requesting MSA application is expecting in reply. In this scenario that the IE904: C_STD_REQ reports a status synchronisation request (the “Status Request Message Type” Data Item in the Status Request IE904 message is “1: Status Synchronisation Request”). The Requested MSA application should reply with the Status Response (IE905: C_STD_RSP) and the missing business (event) message. This is actually the last message sent from the Requesting MSA application.

In particular, the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC of the specific movement in the Requested MSA application. It is the same as the ARC of the request (IE904) message;
- The “Sequence Number”: It includes the latest Sequence Number for the specific ARC in the Requested MSA application;
- The “Status”: It includes the State for the specific ARC in the Requested MSA application;
- The “Last Received Message Type”: It includes
 - “None” if the specific ARC is unknown to the Requested MSA application or the Requested MSA application has *not* received a business (event) message from the Requesting MSA; *OR*

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- the number of the last business (event) message sent from the Requesting MSA application and received by the Requested MSA application for the specific ARC;

In particular, the regenerated “Common Message Header” of the missing business (event) message should include the following items (see Section III.VI.2.1.2):

- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
- The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message (lost/delayed message);
- The “Correlation identifier”: Should equal the value of the “Message identifier” Data Item of the IE904 message. Therefore, it will differ from the value of the original message (lost/delayed message).

The scenario of Manual Status Synchronisation Request for a known ARC is depicted in Figure 122 and Figure 123.

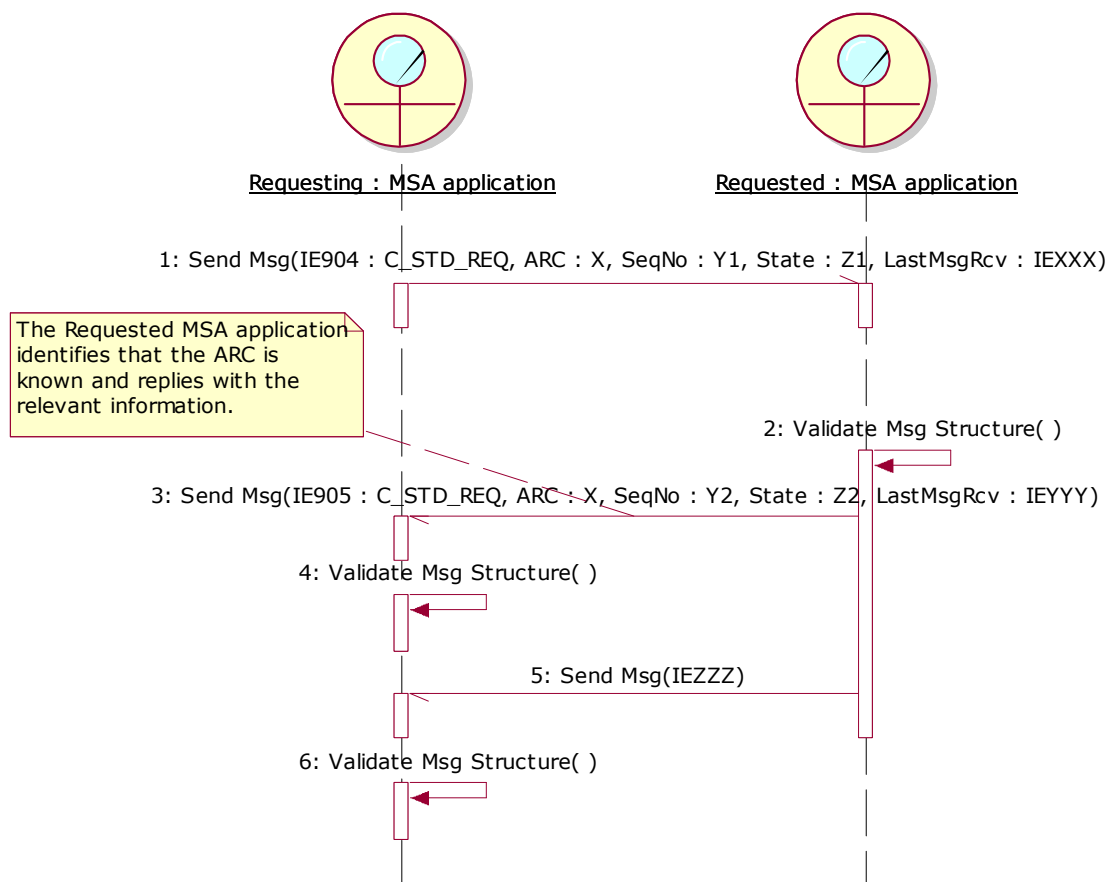


Figure 122: TSD - Manual Status Synchronisation Request for a known ARC

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

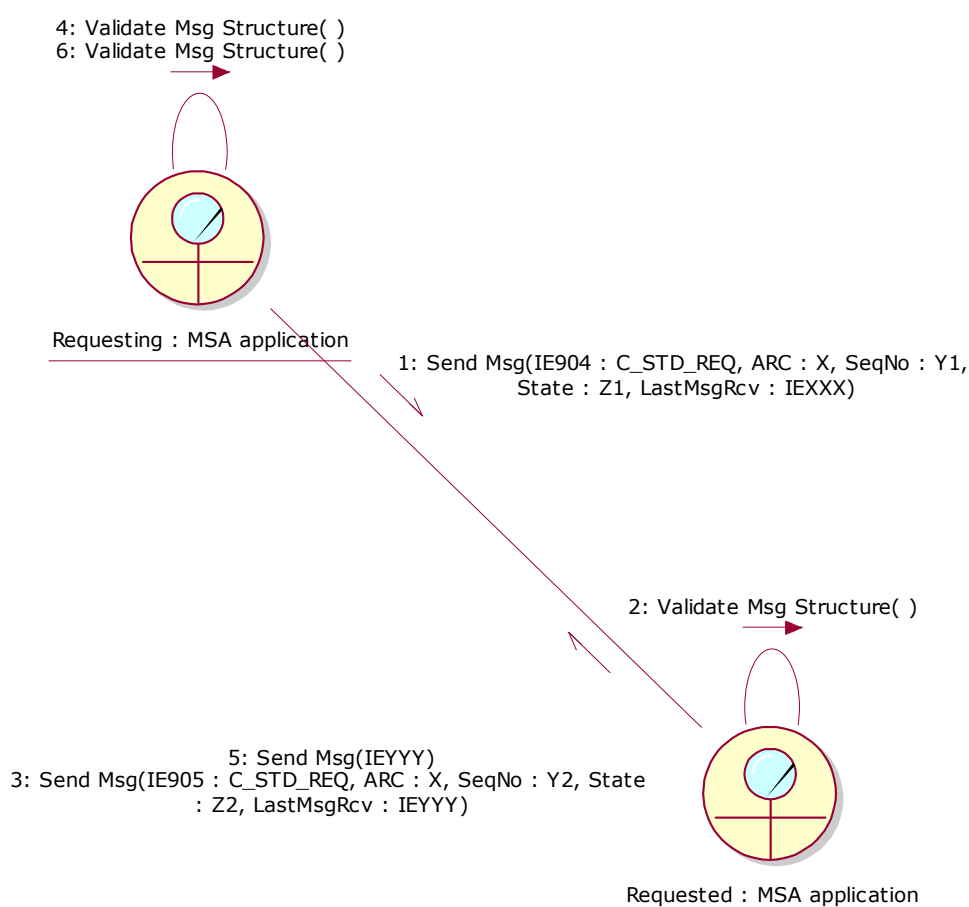


Figure 123: CLD - Manual Status Synchronisation Request for a known ARC

III.VI.2.1.2.2 Manual Status Synchronisation Request for an unknown ARC

The Requested MSA application examines the IE904: C_STD_REQ message content to identify what the Requesting MSA application is expecting in reply. In this scenario that the IE904: C_STD_REQ reports a status synchronisation request (the “Status Request Message Type” Data Item in the Status Request IE904 message is “1: Status Synchronisation Request”). The Requested MSA application should reply with the Status Response (IE905: C_STD_RSP) and the missing business (event) message.

In this scenario the e-AD is unknown to the Requested MSA application, so the Status Response (IE905) should contain “None” as status and “None” as the last business message received from the Requesting MSA application, without sending any subsequent message since no message has been sent to the Requesting MSA application.

Therefore, when an MSA receives a Status Synchronisation Request (IE904) for an unknown e-AD, it should not refuse it with <FUNCTIONAL ERROR.Error Type> equal to error code “Unknown ARC” via an IE906 message.

In particular, the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC provided in the request (IE904) message;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The “Sequence Number”: It includes the value “1”, since the specific ARC is unknown to the Requested MSA application;
- The “Status”: It includes the value “None”, since the specific ARC is unknown to the Requested MSA application.
- The “Last Received Message Type”: It includes the value “None”, since the specific ARC is unknown to the Requested MSA application.

Since the ARC is unknown to the Requested MSA application, there is no message sent from the Requested MSA application to the Requesting MSA application that should be retransmitted.

The scenario of Manual Status Synchronisation Request for an unknown ARC is depicted in Figure 124 and Figure 125:

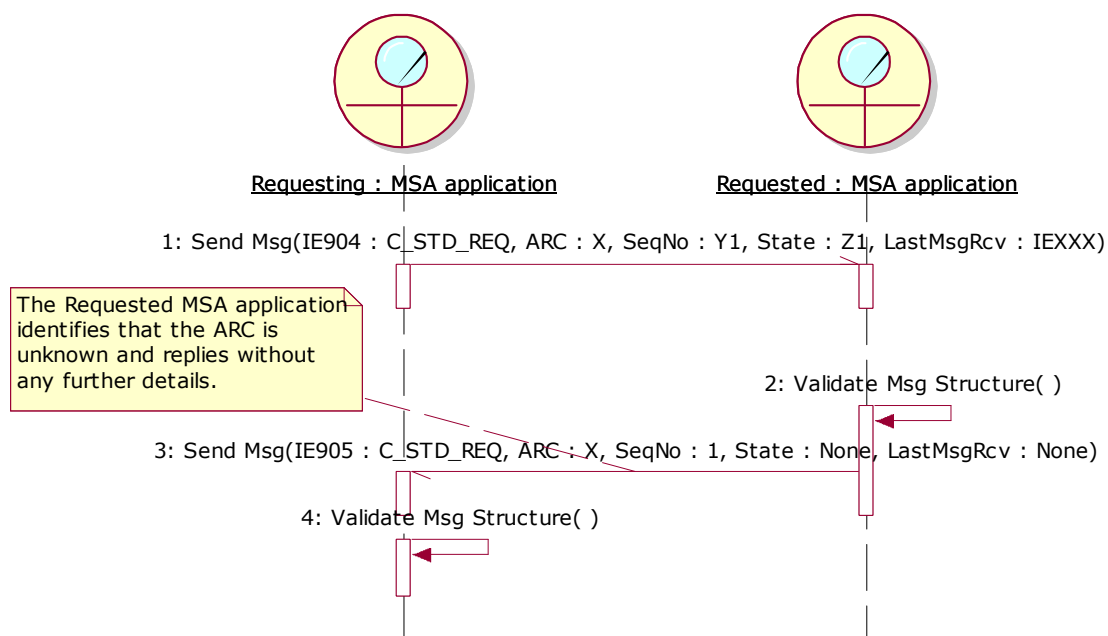


Figure 124: TSD - Manual Status Synchronisation Request for an unknown ARC

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

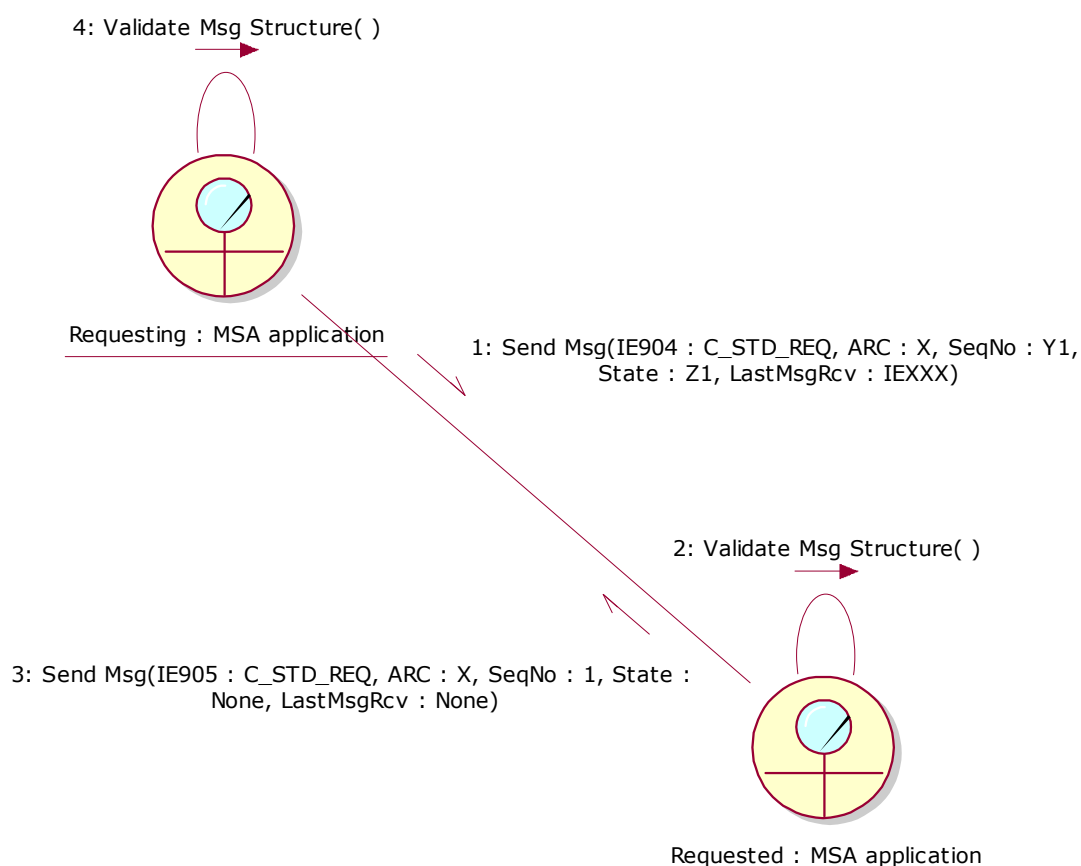


Figure 125: CLD - Manual Status Synchronisation Request for an unknown ARC

III.VI.2.2 Manual Status Request/Response

The Status Request/Response mechanism *could* be used at any time by the MSA dispatch application/MSA destination application to request the status of a particular e-AD at the MSA destination application/MSA dispatch application.

The Status Request (IE904: C_STD_REQ) is triggered manually by the MSA Official for a specific ARC existing in the system. The MSA Official shall indicate that he/she is simply enquiring about the status of the movement. If the IE904: C_STD_REQ sending MSA is the MSA of Dispatch, then the MSA Official shall also provide the sequence number of the last business (event) message sent to the Requested MSA of Destination for the specific ARC. If the IE904: C_STD_REQ sending MSA is the MSA of Destination, then the MSA Official shall also provide the last known sequence number for the specific ARC.

The IE904: C_STD_REQ sending MSA application shall include in the generated Status Request (IE904: C_STD_REQ) message the ARC, the state of the movement as well as the last message received causing a state transition. If the IE904: C_STD_REQ sending MSA is the MSA of Dispatch, then the IE904: C_STD_REQ shall also include the sequence number of the last business (event) message sent to the Requested MSA of Destination for the specific ARC. If the IE904: C_STD_REQ sending MSA is the MSA of Destination, then the IE904: C_STD_REQ shall also include the last known sequence number for the specific ARC.

More specifically the IE904 message should have the following values:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The “Administrative Reference Code”: It includes the ARC provided by the Excise Officer of the Requesting MSA application;
- The “Sequence Number”: It includes the latest Sequence Number for the specific movement in the Requesting MSA application;
- The “Status”: It includes the State for the specific movement in the Requesting MSA application;
- The “Last Received Message Type”: It includes the number of the last business (event) message sent from the Requested MSA application to the Requesting MSA application for the specific ARC;
- The “Status Request Message Type”: This item is not applicable in case the IE904 message is sent for a Status Request/Response scenario.

On the other side, the IE904: C_STD_REQ receiving MSA examines the message content to identify what the sending MSA is expecting in reply. In this scenario that the IE904: C_STD_REQ reports a simple status request, the IE904: C_STD_REQ receiving MSA replies with a single message, the Status Response (IE905: C_STD_RSP). The Status Response (IE905: C_STD_RSP) message shall include the current status of the movement as well as the last received business (event) message from the IE904: C_STD_REQ sending MSA. If the Requested MSA is the MSA of Dispatch, then the Status Response (IE905: C_STD_RSP) shall also include the sequence number of the last business (event) message sent to the MSA of Destination for the specific ARC. If the Requested MSA is the MSA of Destination, then the Status Response (IE905: C_STD_RSP) shall also include the last known sequence number for the specific ARC.

More specifically the IE905 message should have the following values:

- The “Administrative Reference Code”: It includes the ARC of the specific movement in the Requested MSA application. It is the same as the ARC of the request (IE904) message;
- The “Sequence Number”: It includes the latest Sequence Number for the specific ARC in the Requesting MSA application;
- The “Status”: It includes the State for the specific ARC in the Requesting MSA application;
- The “Last Received Message Type”: It includes the number of the last business (event) message sent from the Requested MSA application to the Requesting MSA application.

The figures below (Figure 126 and Figure 127) illustrate an indicative example of the MSA dispatch application requesting the status of an e-AD, which is found in the “*Accepted*” state. In that request, the MSA destination application responds with an IE905: C_STD_RSP indicating that e-AD is also found in the “*Accepted*” state.

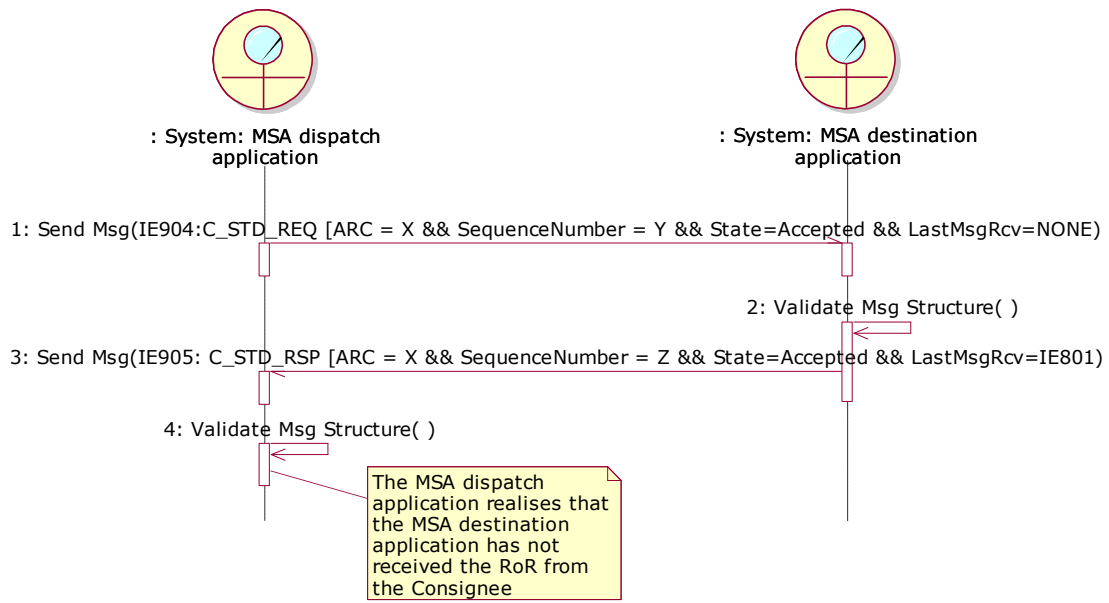


Figure 126: TSD - Manual Status Request/Response

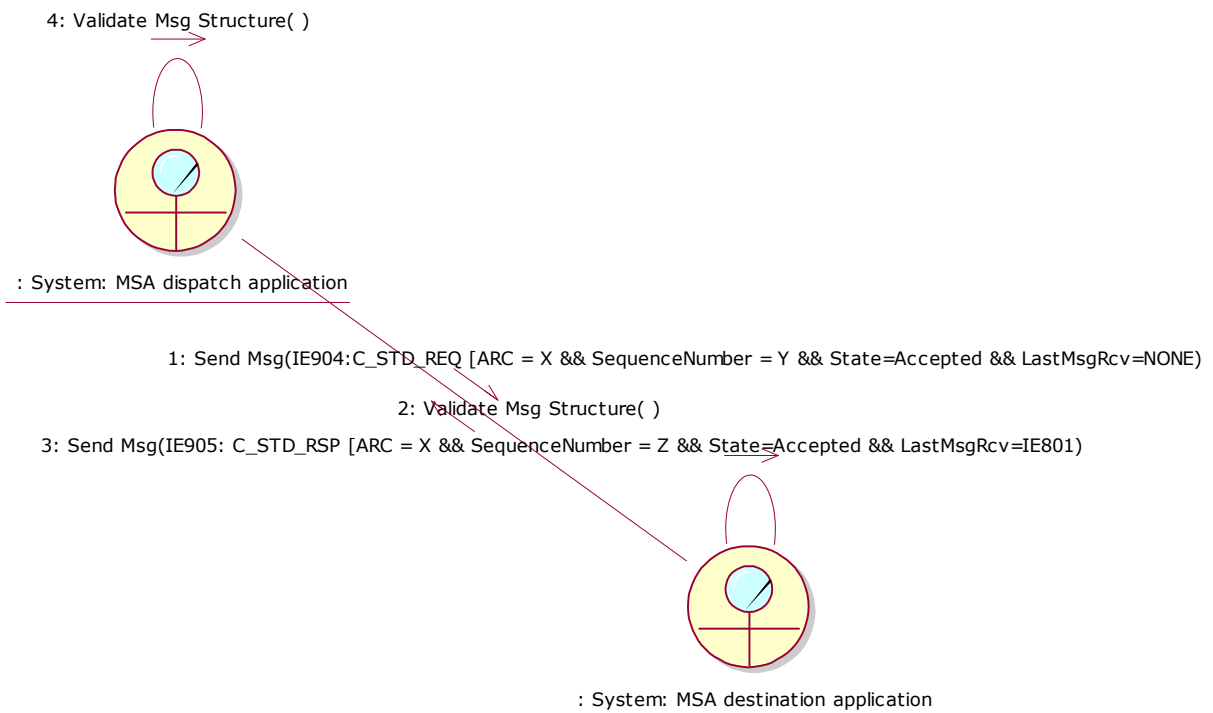


Figure 127: CLD - Manual Status Request/Response

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.3 Manual Status Synchronisation Request

In Section III.VI.2.2 Manual Status Request/Response, the MSA Official issues a simple status request, reporting that he/she wishes to receive in reply only the movement status at the communicating MSA. This section presents the “Manual Status Synchronisation Request” mechanism, which *could* be provided by a MSA to its MSA Officials as a more advanced (sophisticated) use of the “Manual Status Request/Response”. In particular, the “Manual Status Synchronisation Request” *could* be used at any time by the MSA Officials to request a movement status synchronisation by receiving in reply of a status request not only the status of the movement but also any missing/delayed messages.

The following scenarios have been identified:

III.VI.2.3.1 Submission of RoR and the e-AD is under the ‘Accepted’ state at the MSA of Dispatch

According to the following scenario, the RoR (IE818: C_DEL_DAT) has been submitted by the MSA of Destination. However, the IE818: C_DEL_DAT message has not been received by the MSA of Dispatch and the e-AD is still in the “Accepted” state.

The Status Request (IE904: C_STD_REQ) message is triggered manually by the Official at the MSA of Dispatch. The MSA Official shall provide the ARC of the specific movement accompanied by the sequence number of the last business (event) message sent to the MSA of Destination for the specific ARC. In addition, the MSA Official shall indicate that he/she is also wishing to synchronise the movement state in case a state de-synchronisation is detected.

The MSA dispatch application shall include in the generated Status Request (IE904: C_STD_REQ) message the sequence number of the last business (event) message sent to the MSA of Destination for the specific ARC, the state of the movement (“Accepted”) and the last business (event) message received from the MSA destination application (None). In addition, the MSA dispatch application has to declare in the sending IE904: C_STD_REQ that it requests a Status Synchronisation Request. If the latter is not indicated then only the status of the movement will be returned back from the initiating MSA.

Upon receipt of the Status Request (IE904: C_STD_REQ) message, the MSA destination application examines its content.

The MSA destination application identifies that:

- The MSA dispatch application is still in the “Accepted” state, while the RoR (the IE818: C_DEL_DAT) has already been communicated to it;
- The “Message Type” in the Status Request (IE904: C_STD_REQ) message is “1: Status Synchronisation Request”, indicating that the MSA Official at the MSA of Dispatch wishes also to receive the missing IE818: C_DEL_DAT message in order to synchronise the movement status.

The MSA destination application sends to the MSA dispatch application:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The Status Response (IE905: C_STD_RSP) message with the status set to “Delivered” or “Partially Refused” or “Refused” and the information that the last received message from the MSA dispatch application is “IE801: C_EAD_VAL” as well as the last known sequence number for the specific ARC, which would be the same with that included in the RoR, followed by;
- The RoR (IE818: C_DEL_DAT). The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2). Due to the correlation mechanism that is presented in the introduction of Section III.VI.2.1, the “Correlation Identifier” Data Item of the IE905 and the regenerated IE818 message should be equal to the “Message identifier” Data Item of the IE904 message:
 - The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
 - The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message;
 - The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message.

Upon the reception of a valid RoR (IE818: C_DEL_DAT), the MSA dispatch application validates successfully its structure, stores its data and changes the state of the e-AD from “Accepted” to “Delivered” or to “Partially Refused” or to “Refused”.

Finally, it is recommended that the MSA dispatch application forwards the RoR (IE818: C_DEL_DAT) to the Consignor.

This scenario is depicted in Figure 128 and Figure 129:

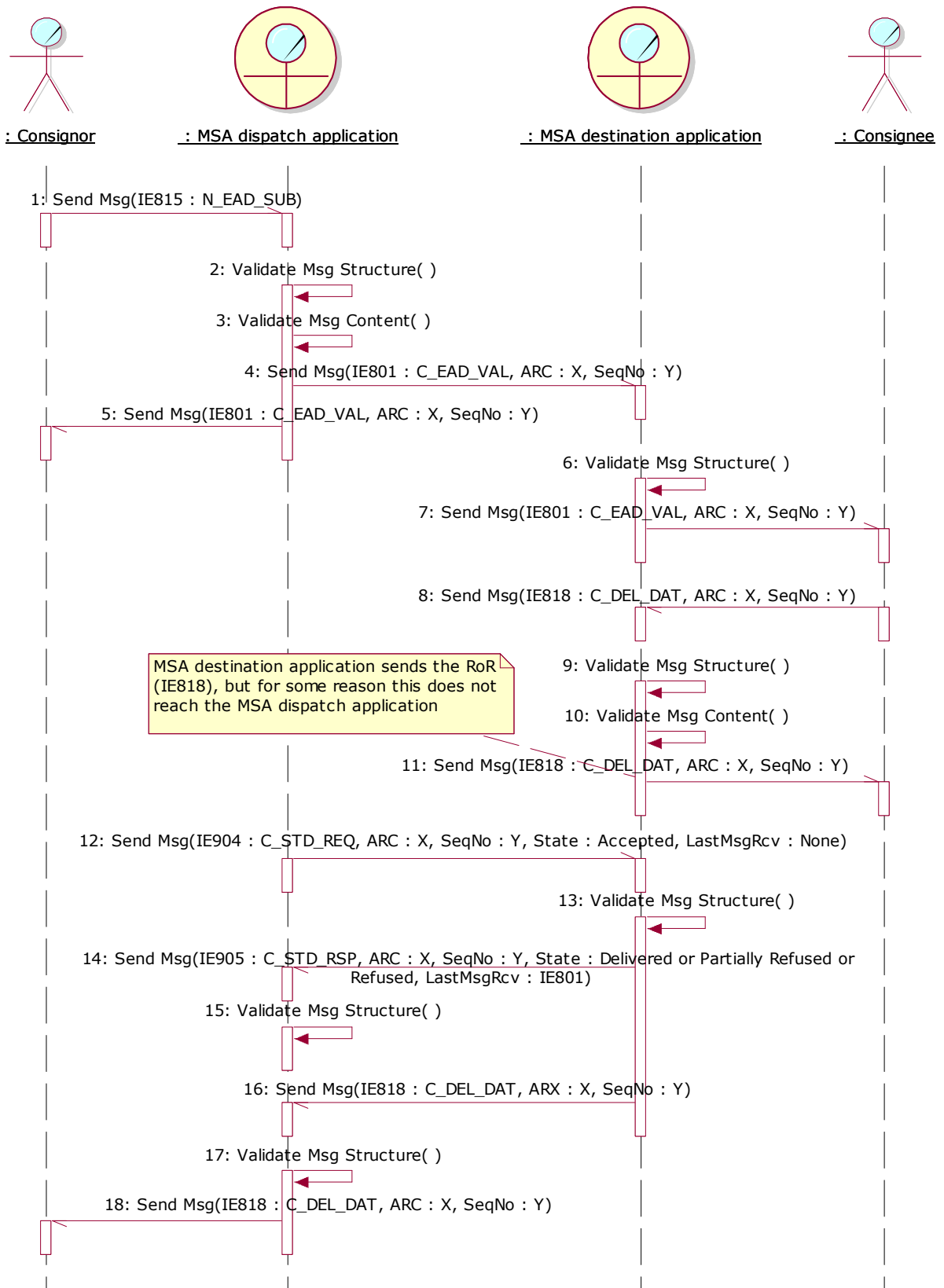


Figure 128: TSD - Status Synchronisation Request for a missing RoR (IE818) - Re-submission of RoR (IE818)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

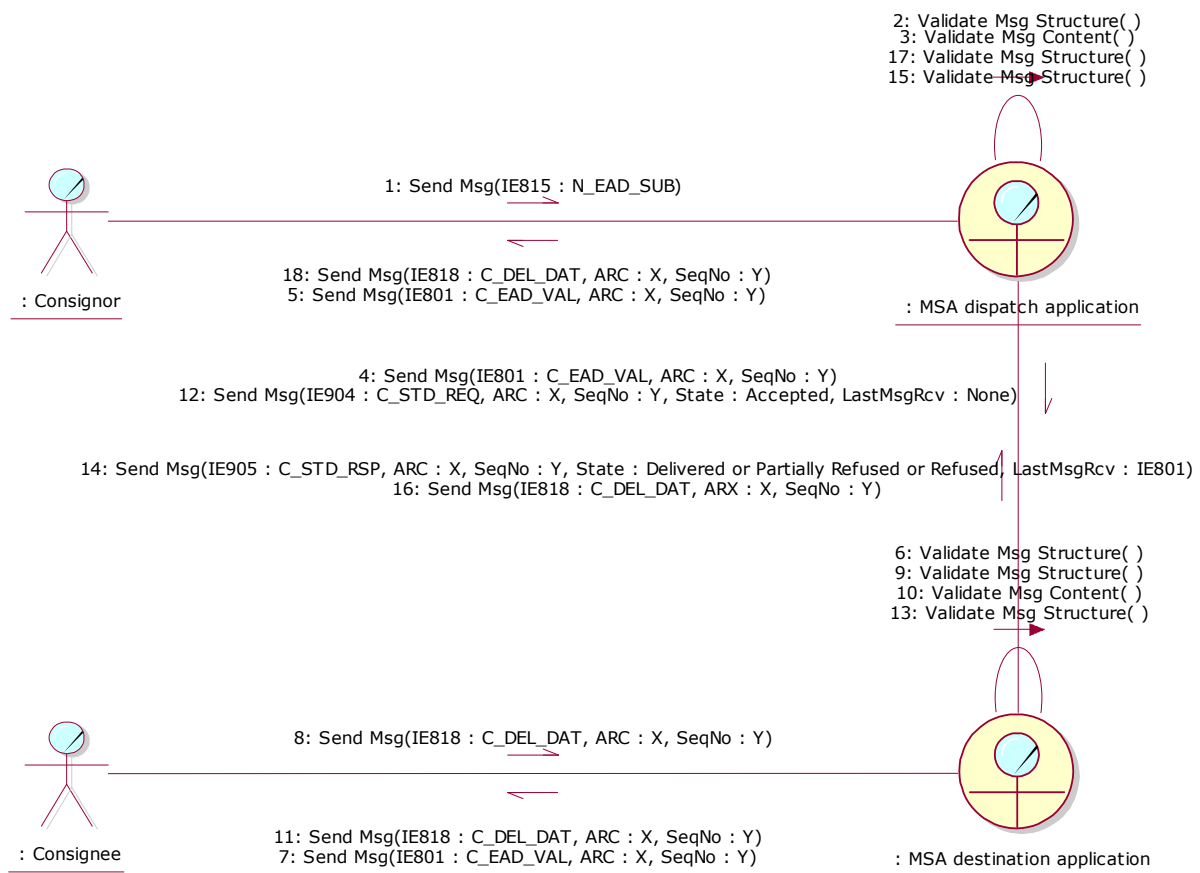


Figure 129: CLD - Status Synchronisation Request for a missing RoR (IE818) - Re-submission of RoR (IE818)

III.VI.2.3.2 Change of MSA of Destination and the e-AD is under the ‘Accepted’, or ‘Partially Refused’ or ‘Refused’ state at the MSA of Destination

According to the following scenario, a change of MSA of Destination has occurred. However, the IE813: C_UPD_DAT has not been received by the former MSA of Destination and the e-AD is still in the “Accepted” or “Partially Refused” or “Refused” state.

The Status Request (IE904: C_STD_REQ) message is triggered manually by the Official at the MSA of Destination. The MSA Official shall provide the ARC of the specific movement accompanied by the last known sequence number for the specific ARC. In addition, the MSA Official shall indicate that he/she is also wishing to synchronise the movement state in case a state de-synchronisation is detected.

The MSA destination application shall include in the generated Status Request (IE904: C_STD_REQ) message the last known sequence number for the specific ARC, the state of the movement (“Accepted” or “Partially Refused” or “Refused”) and the last business (event) message received from the MSA dispatch application (IE801: C_EAD_VAL). In addition, the MSA dispatch application has to declare in the sending IE904: C_STD_REQ that it requests a Status Synchronisation Request. If the latter is not indicated then only the status of the movement will be returned back from the initiating MSA.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon receipt of the Status Request (IE904: C_STD_REQ) message, the MSA dispatch application examines its content.

The MSA dispatch application identifies that:

- The MSA destination application is still in the “*Accepted*” or “*Partially Refused*” or “*Refused*” state, while an update message (IE813: C_UPD_DAT) has already been communicated to it;
- The “Message Type” in the Status Request (IE904: C_STD_REQ) message is “1: Status Synchronisation Request”, indicating that the MSA Official at the MSA of Destination wishes also to receive the missing IE813: C_UPD_DAT message in order to synchronise the movement status.

The MSA dispatch application sends to the MSA destination application:

- The Status Response (IE905: C_STD_RSP) message with the sequence number set to that of the last business (event) message sent to the MSA of Destination, the status set to “*Accepted*” or “*Delivered*” or “*Partially Refused*” or “*Refused*” and the information that the last received message from the MSA destination application is “*None*”, when the state in the MSA dispatch application is “*Accepted*”, or that the last received message from the MSA destination application is RoR (IE818: C_DEL_DAT), when the state is “*Delivered*”, “*Refused*” or “*Partially Refused*”, followed by;
- The update message (IE813: C_UPD_DAT). The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2). Due to the correlation mechanism that is presented in the introduction of Section III.VI.2.1, the “Correlation Identifier” Data Item of the IE905 and the regenerated IE813 message should be equal to the “Message identifier” Data Item of the IE904 message;
- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
- The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message;
- The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message.

Upon the reception of a valid update message (IE813: C_UPD_DAT), the MSA destination application validates successfully the structure of the update message, stores the updated information and changes the state of the e-AD from “*Accepted*” or “*Refused*” to “*Diverted*” or from “*Partially Refused*” to “*Delivered*”, which are final states.

Finally, it is recommended that the MSA destination application forwards the notification message (IE803: C_EAD_NOT) to the former Consignee.

This scenario is depicted in Figure 130 and Figure 131:

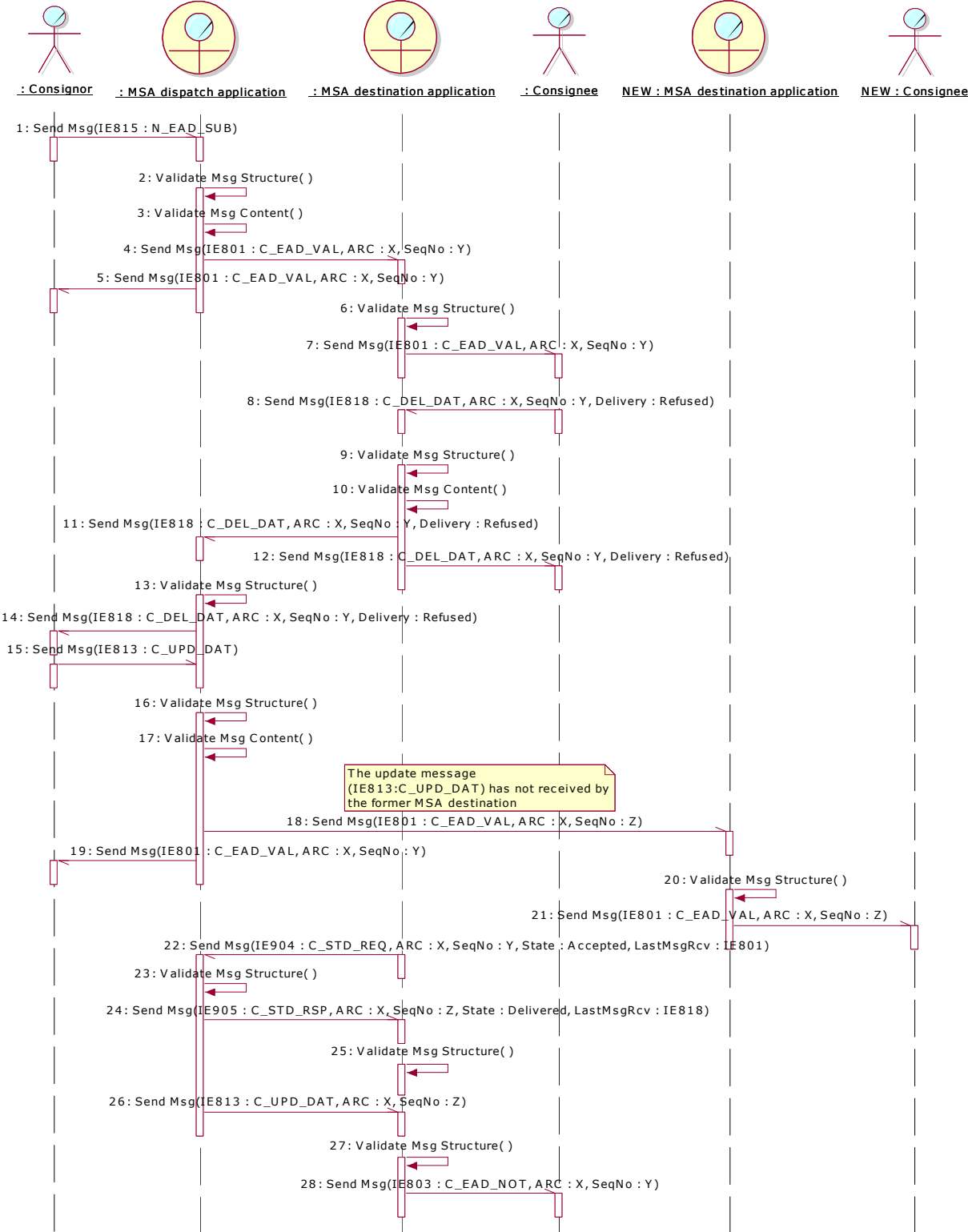


Figure 130: TSD - Status Synchronisation Request for a missing Update Message (IE813) - Re-submission of Update Message (IE813)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

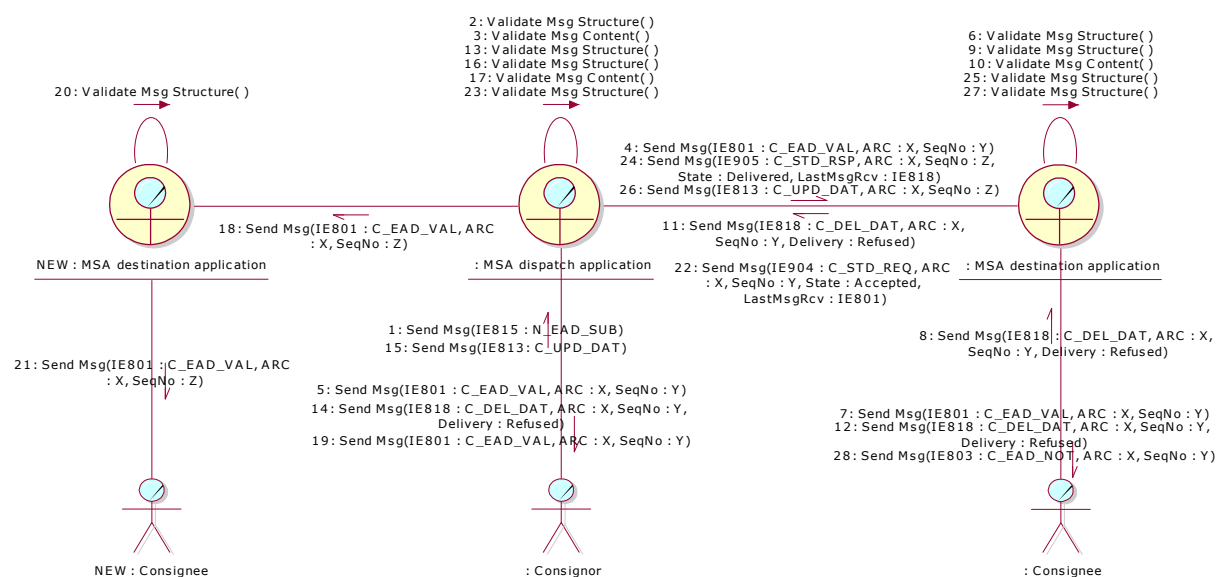


Figure 131: CLD - Status Synchronisation Request for a missing Update Message (IE813) - Re-submission of Update Message (IE813)

III.VI.2.3.3 Cancellation and the e-AD is under the ‘Accepted’ state at the MSA of Destination

According to the following scenario, a movement has been cancelled at the MSA of Dispatch (e-AD state at the MSA of Dispatch = “Cancelled”). However, the cancellation notification (IE810: C_CAN_DAT) has not been properly received by the MSA of Destination and the e-AD is still in the “Accepted” state.

The Status Request (IE904: C_STD_REQ) message is triggered manually by the Official at the MSA of Destination. The MSA Official shall provide the ARC of the specific movement accompanied by the last known sequence number for the specific ARC. In addition, the MSA Official shall indicate that he/she is also wishing to synchronise the movement state in case a state de-synchronisation is detected.

The MSA destination application shall include in the generated Status Request (IE904: C_STD_REQ) message the last known sequence number for the specific ARC, the state of the movement (“Accepted”) as well as the last business (event) message received from the MSA dispatch application (IE801: C_EAD_VAL). In addition, the MSA dispatch application has to declare in the sending IE904: C_STD_REQ that it requests a Status Synchronisation Request. If the latter is not indicated then only the status of the movement will be returned back from the initiating MSA.

Upon receipt of the Status Request (IE904: C_STD_REQ) message, the MSA dispatch application examines its content.

The MSA dispatch application identifies that:

- The MSA destination application is still in the “Accepted” state, while a cancellation notification (IE810: C_CAN_DAT) has already been communicated to it;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The “Message Type” in the Status Request (IE904: C_STD_REQ) message is “1: Status Synchronisation Request”, indicating that the MSA Official at the MSA of Destination wishes also to receive the missing IE810: C_CAN_DAT message in order to synchronise the movement status.

The MSA dispatch application sends to the MSA destination application:

- The Status Response (IE905: C_STD_RSP) message with the sequence number set to that of the last business (event) message sent to the MSA of Destination, the status set to “*Cancelled*” and the information that the last received message from the MSA destination application is “None”, followed by;
- The cancellation notification message (IE810: C_CAN_DAT). The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2). Due to the correlation mechanism that is presented in the introduction of Section III.VI.2.1, the “Correlation Identifier” Data Item of the IE905 and the regenerated IE810 message should be equal to the “Message identifier” Data Item of the IE904 message:
- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
- The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message;
- The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message.

Upon the reception of a valid cancellation (IE810: C_CAN_DAT), the MSA destination application validates successfully the structure of the cancellation message, stores the cancellation information and changes the state of the e-AD from “*Accepted*” to “*Cancelled*”, which is a final state.

Finally, it is recommended that the MSA destination application forwards the cancellation notification (IE810: C_CAN_DAT) to the Consignee.

This scenario is depicted in Figure 132 and Figure 133:

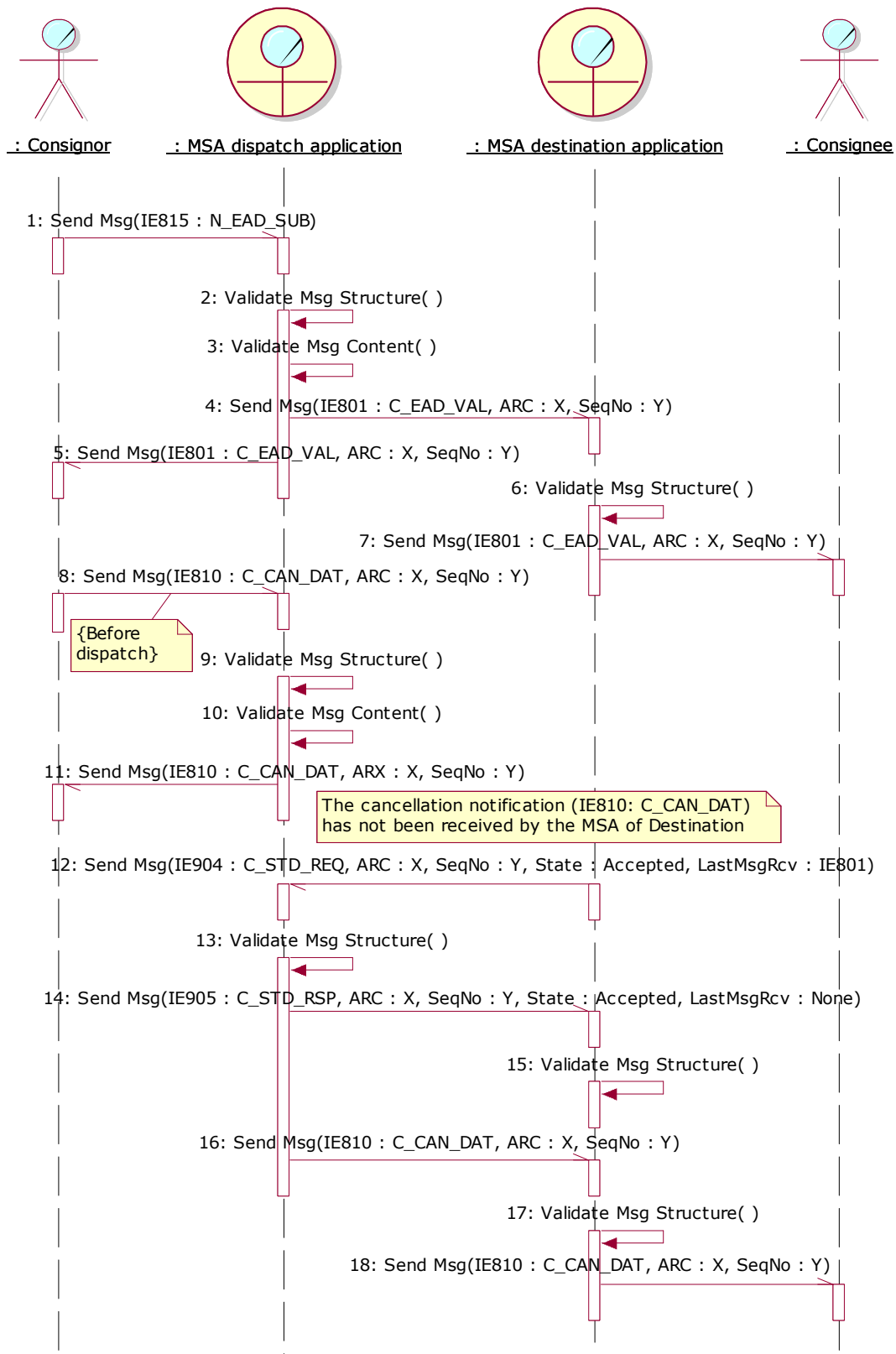


Figure 132: TSD - Status Synchronisation Request for a missing Cancellation Notification (IE810) - Re-submission of Cancellation Notification (IE810)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

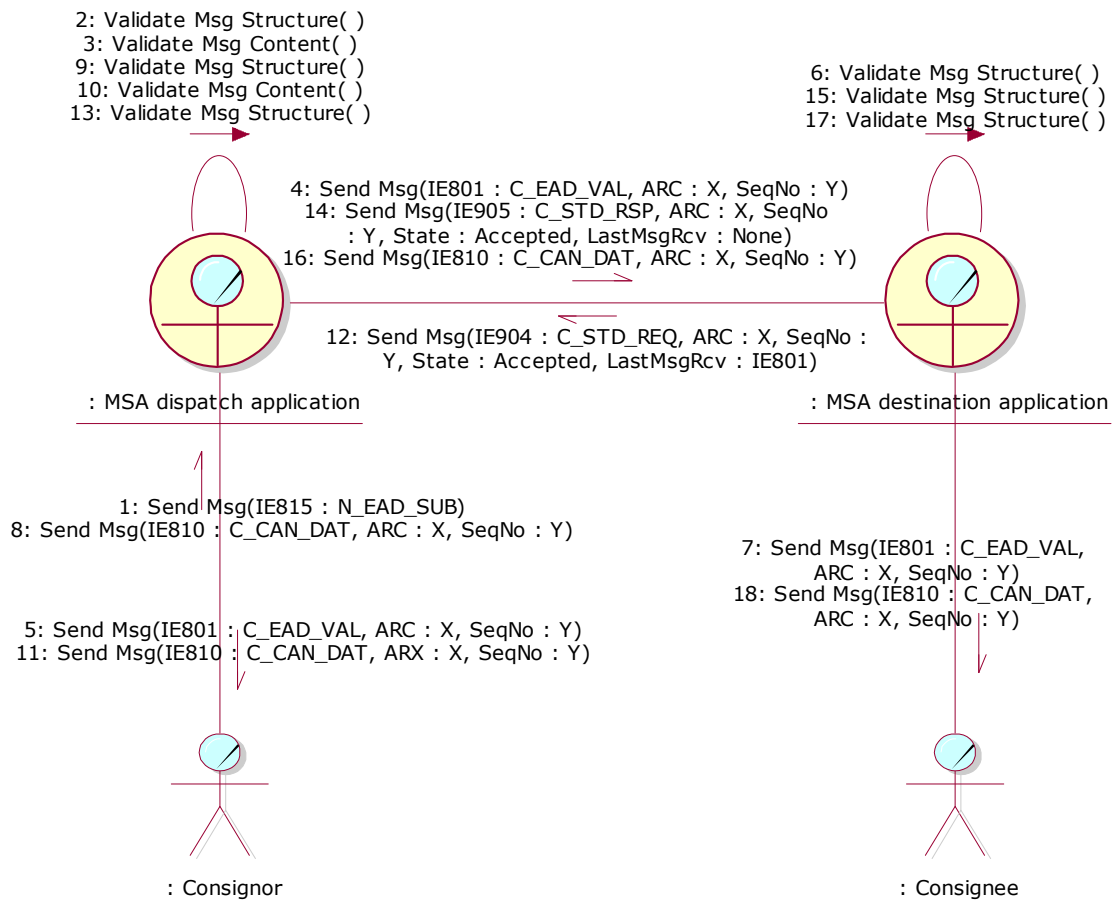


Figure 133: CLD - Status Synchronisation Request for a missing Cancellation Notification (IE810) - Re-submission of Cancellation Notification (IE810)

III.VI.2.3.4 e-AD Manual Closure and the e-AD is under the ‘Accepted’ state at the MSA of Destination

According to the following scenario, the movement has been manually closed at the MSA of Dispatch (e-AD state at the MSA of Dispatch = “e-AD Manually Closed”). However, the Status Response (IE905: C_STD_RSP) has not properly received by the MSA of Destination and the e-AD is still in the “Accepted” state (please, also refer to Section III.VI.3 Manual Closing of the Movement).

The Status Request (IE904: C_STD_REQ) message is triggered manually by the Official at the MSA of Destination. The MSA Official shall provide the ARC of the specific movement accompanied by the last known sequence number for the specific ARC. In addition, the MSA Official shall indicate that he/she is also wishing to synchronise the movement state in case a state de-synchronisation is detected.

The MSA destination application shall include in the generated Status Request (IE904: C_STD_REQ) message the last known sequence number for the specific ARC, the state of the movement (“Accepted”) and the last business (event) message received from the MSA dispatch application (IE801: C_EAD_VAL). In addition, the MSA dispatch application has to declare in the sending IE904: C_STD_REQ that it requests a Status Synchronisation Request.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Upon receipt of the Status Request (IE904: C_STD_REQ) message, the MSA dispatch application examines its content.

The MSA dispatch application identifies that:

- The MSA destination application is still in the “*Accepted*” state, while a manual closure notification (IE905: C_STD_RSP) has already been communicated to it;
- The “Message Type” in the Status Request (IE904: C_STD_REQ) message is “1: Status Synchronisation Request”, indicating that the MSA Official at the MSA of Destination wishes also to receive the missing IE905: C_STD_RSP message in order to synchronise the movement status.

The MSA dispatch application sends to the MSA destination application:

- The Status Response (IE905: C_STD_RSP) message with the sequence number set to that of the last business (event) message sent to the MSA of Destination, the status set to “*e-AD Manually Closed*” and the information that the last received message from the MSA destination application is “*None*”. Due to the correlation mechanism that is presented in the introduction of Section III.VI.2.1, the “Correlation Identifier” Data Item of the IE905 should be equal to the “Message identifier” Data Item of the IE904 message:
- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
- The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message;
- The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message.

As, in this case, the missing message is the IE905: C_STD_RSP, the MSA dispatch application shall not send it twice to the MSA destination application.

Upon the reception of a valid IE905: C_STD_RSP message, the MSA destination application validates successfully its structure, stores its information and changes the state of the e-AD from “*Accepted*” to “*e-AD Manually Closed*”, which is a final state.

This scenario is depicted in Figure 134 and Figure 135:

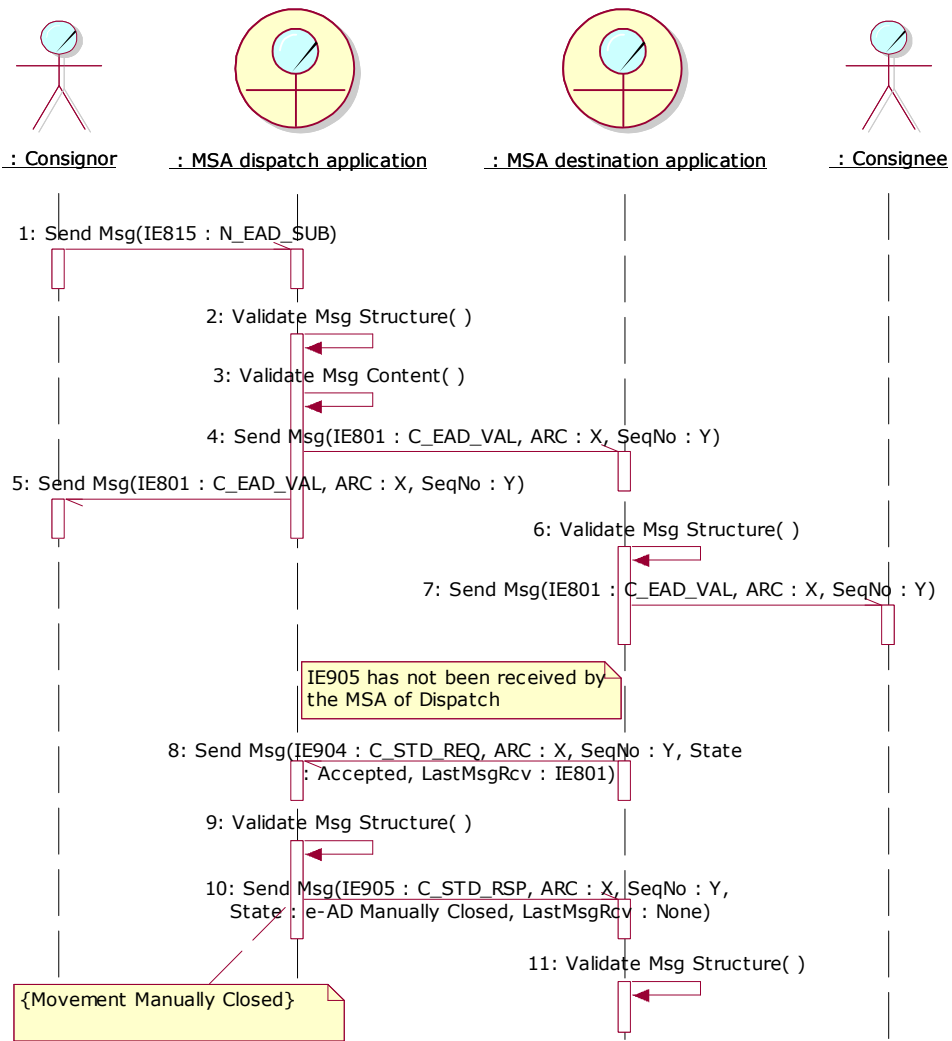


Figure 134: TSD - Status Synchronisation Request for a missing Manual Closure Notification (IE905)

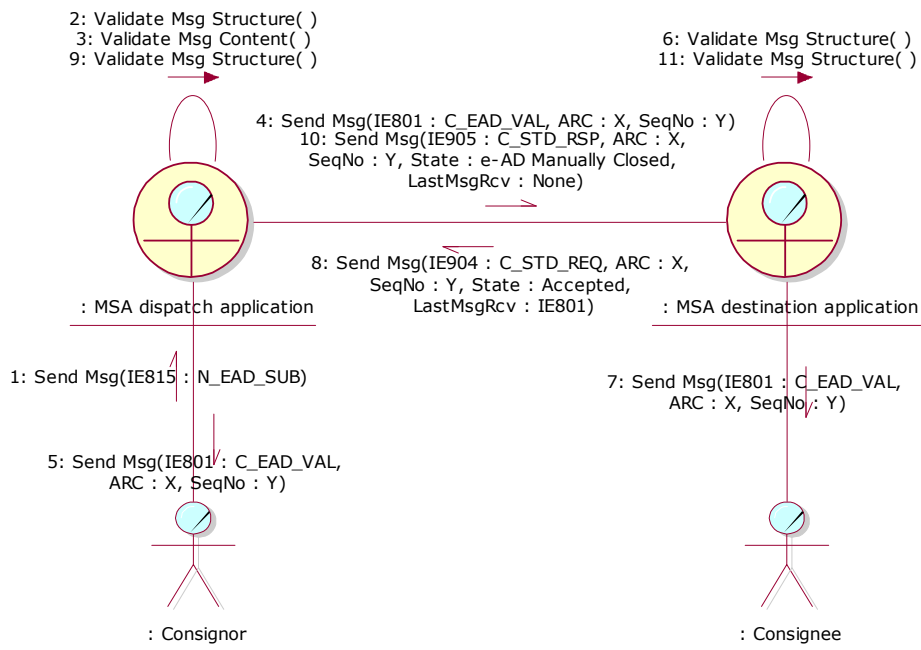


Figure 135: CLD - Status Synchronisation Request for a missing Manual Closure Notification (IE905)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.3.5 Submission of e-AD rejection and the e-AD is under the ‘Accepted’ state at the MSA of Dispatch

According to the following scenario, a movement has been rejected at the MSA of Destination (e-AD state at the MSA of Destination = “*Rejected*”). However, the rejection notification (IE819: C_REJ_DAT) has not been properly received by the MSA of Dispatch and the e-AD is still in the “*Accepted*” state.

The Status Request (IE904: C_STD_REQ) message is triggered manually by the Official at the MSA of Dispatch. The MSA Official shall provide the ARC of the specific movement accompanied by the last known sequence number for the specific ARC. In addition, the MSA Official shall indicate that he/she is also wishing to synchronise the movement state in case a state de-synchronisation is detected.

The MSA dispatch application shall include in the generated Status Request (IE904: C_STD_REQ) message the last known sequence number for the specific ARC, the state of the movement (“*Accepted*”) as well as the last business (event) message received from the MSA destination application (“*None*”). In addition, the MSA dispatch application has to declare in the sending IE904: C_STD_REQ that it requests a Status Synchronisation Request. If the latter is not indicated then only the status of the movement will be returned back from the initiating MSA.

Upon receipt of the Status Request (IE904: C_STD_REQ) message, the MSA destination application examines its content.

The MSA destination application identifies that:

- The MSA dispatch application is still in the “*Accepted*” state, while a rejection notification (IE819: C_REJ_DAT) has already been communicated to it;
- The “Message Type” in the Status Request (IE904: C_STD_REQ) message is “1: Status Synchronisation Request”, indicating that the MSA Official at the MSA of Dispatch wishes also to receive the missing IE819: C_REJ_DAT message in order to synchronise the movement status.

The MSA destination application sends to the MSA dispatch application:

- The Status Response (IE905: C_STD_RSP) message with the sequence number set to that of the last business (event) message sent to the MSA of Destination, the status set to “*Rejected*” and the information that the last received message from the MSA destination application is the e-AD (IE801: C_EAD_VAL), followed by;
- The rejection notification message (IE819: C_REJ_DAT). The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2). Due to the correlation mechanism that is presented in the introduction of Section III.VI.2.1, the “Correlation Identifier” Data Item of the IE905 and the regenerated IE819 message should be equal to the “Message identifier” Data Item of the IE904 message:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

- The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
- The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message;
- The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message.

Upon the reception of a valid rejection (IE819: C_REJ_DAT), the MSA dispatch application validates successfully the structure of the rejection message, stores the rejection information and changes the state of the e-AD from “*Accepted*” to “*Rejected*”, which is a final state.

Finally, it is recommended that the MSA dispatch application forwards the rejection notification (IE819: C_REJ_DAT) to the Consignor.

This scenario is depicted in Figure 136 and Figure 137:

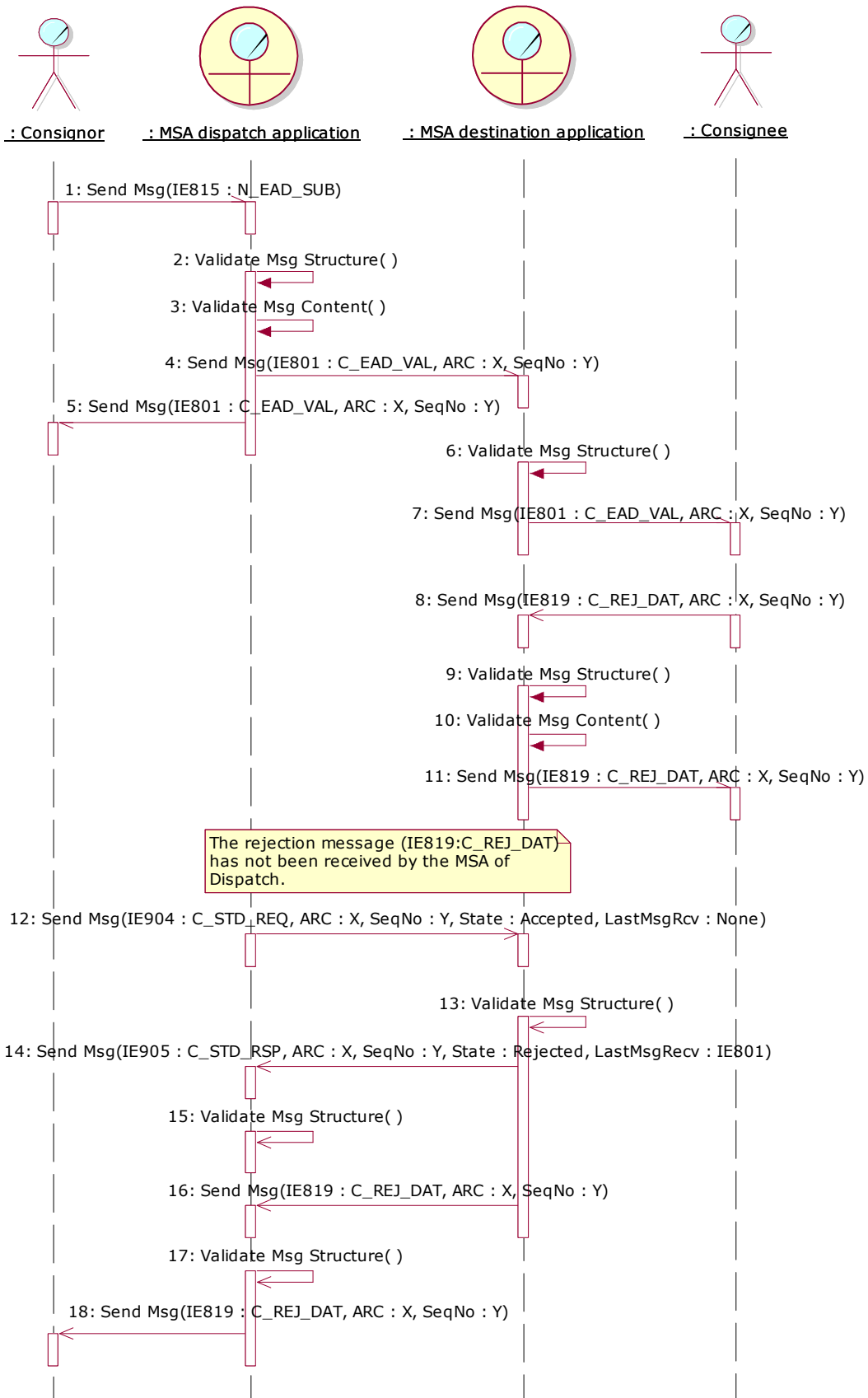


Figure 136: TSD - Status Synchronisation Request for a missing Rejection Notification (IE819) - Re-submission of Rejection Notification (IE819)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

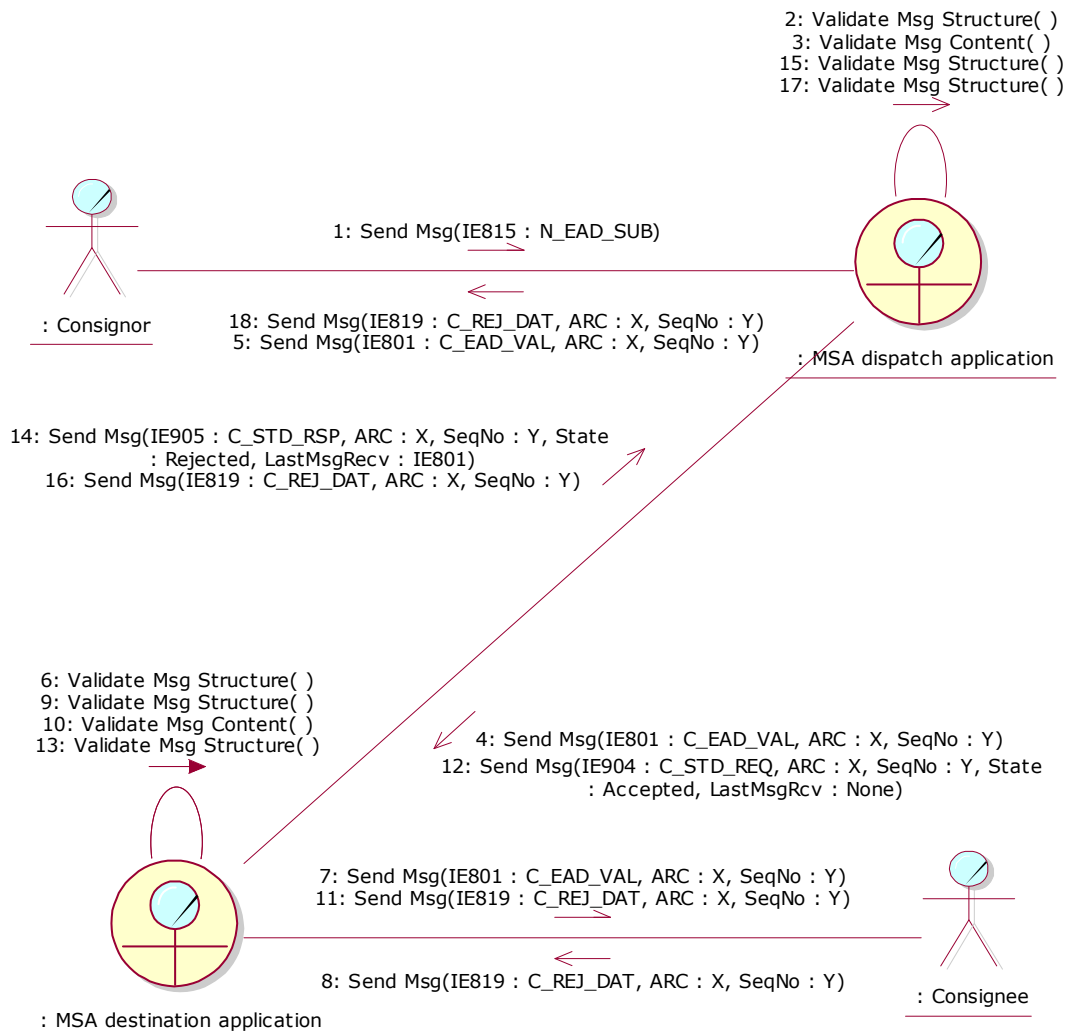


Figure 137: CLD - Status Synchronisation Request for a missing Rejection Notification (IE819) - Re-submission of Rejection Notification (IE819)

III.VI.2.4 Automatic Status Synchronisation Request

A MSA *shall* use the IE904/IE905 mechanism to handle exceptions. In particular, the IE904/IE905 mechanism shall be automatically triggered in order to identify whether an e-AD/RoR is missing due to a technical failure or due to a business discrepancy.

III.VI.2.4.1 Receipt of RoR for Unknown ARC

The basic scenario describes the procedure of submission of e-AD (Chapter III.I.1). According to that scenario, both the MSA dispatch application and the MSA destination application register successfully the new e-AD.

In particular, there is an exceptional case, where the MSA destination application may not receive the e-AD (IE801: C_EAD_VAL) from the MSA dispatch application (possibly due to a network failure). Hence, an issue is raised when the movement physically arrives at Destination on time and the Consignee wants to send the RoR (IE818: C_DEL_DAT). In that case, the MSA destination application shall be able to automatically request the e-AD information from the MSA dispatch application.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

Hence, the Consignee sends the RoR for a specific ARC, which is unknown at the MSA of Destination application. Then, the MSA destination application automatically generates and sends to the MSA dispatch application a Status Request (IE904: C_STD_REQ) for the ARC and sequence number included in the received RoR (sent by the Consignee), including also the information that the status is “None” and no message has been received from the MSA dispatch application. In addition, the indication of a Status Synchronisation Request in the IE904 will be automatically set by the application.

Upon the reception of the Status Request (IE904: C_STD_REQ), the MSA dispatch application validates the request and checks if the ARC is known.

The following two scenarios may be encountered:

III.VI.2.4.1.1 Missed e-AD

The MSA dispatch application detects that the requested ARC is known and that it is in the “Accepted” state. It also detects that the “Message Type” in the IE904 message is “1: Status Synchronisation Request”, meaning that the IE904 message was submitted for state synchronisation purposes. In this case, the MSA dispatch application responds by sending:

- The Status Response (IE905: C_STD_RSP) with the sequence number set to that of the last business (event) message sent to the MSA of Destination, the status set to “Accepted” and the information that the last message received from the MSA destination application is “None”;
- The missing e-AD (IE801: C_EAD_VAL) to the MSA destination application. The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2):
 - The “Message sender”, “Message recipient”, “Date of preparation” and “Time of preparation” Data Items. The values of the aforementioned Data Items of the regenerated message will have the same values as the original message;
 - The “Message identifier”: The “Message Identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message (lost/delayed message);
 - The “Correlation identifier”: It does not apply for requests and one way messages. The “Correlation identifier” value of the message sent due to Status Synchronisation Request will differ from the value of the original message (lost/delayed message).

The received e-AD is stored by the MSA destination application and the movement state is set to “Accepted”.

Moreover, the MSA destination application proceeds with the successful validation of the draft RoR, the message exchanges and the state transitions as the scenario in Section III.I.4 defines.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

If the MSA destination application also receives the original e-AD (IE801: C_EAD_VAL), either before or after the reception of the re-submitted e-AD (IE801: C_EAD_VAL), the application shall behave in accordance with the requirements in Section III.VI.2.1.2 Manual Status Synchronisation Request.

This scenario is depicted in Figure 138 and Figure 139:

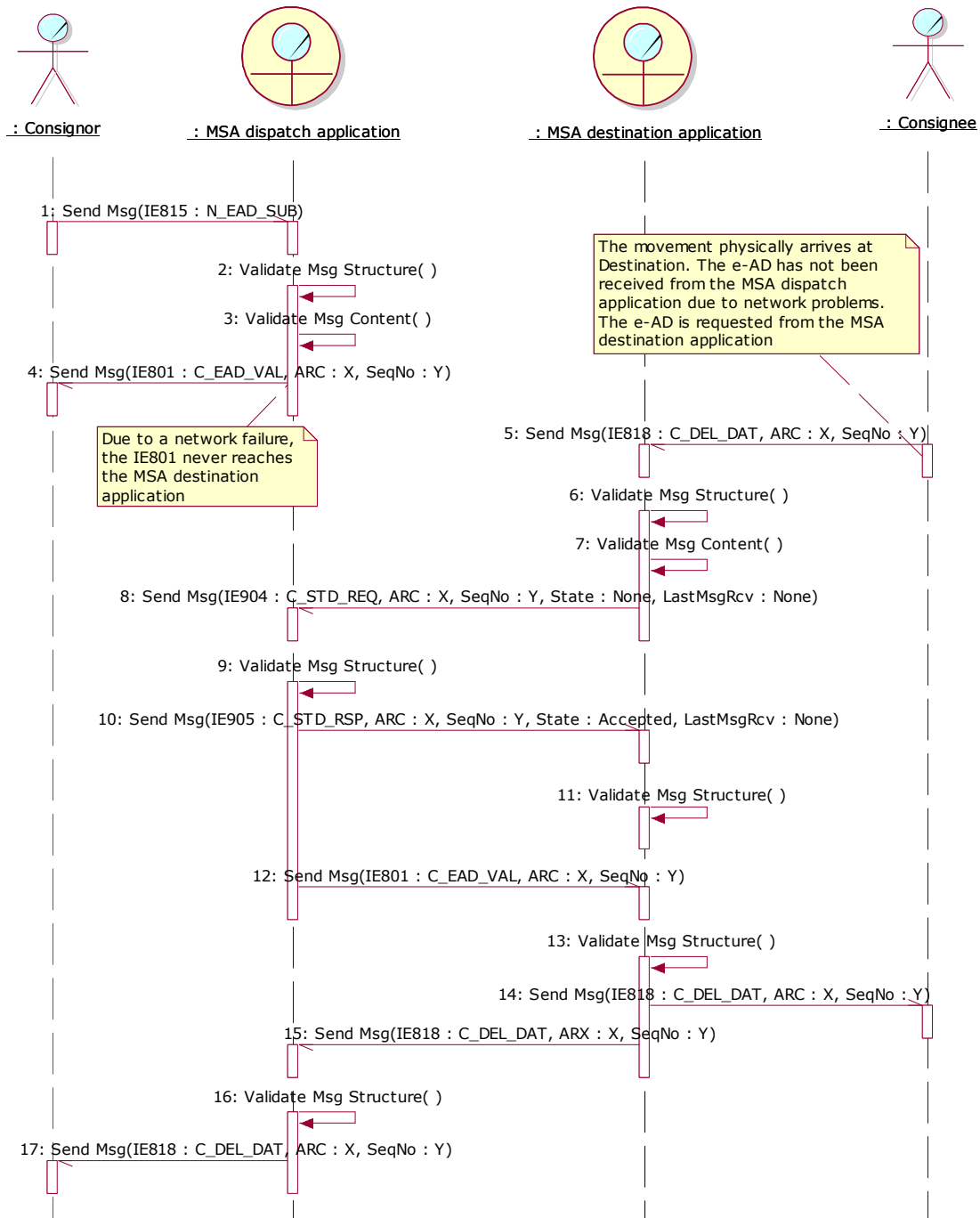


Figure 138: TSD - Positive response on a requested e-AD

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

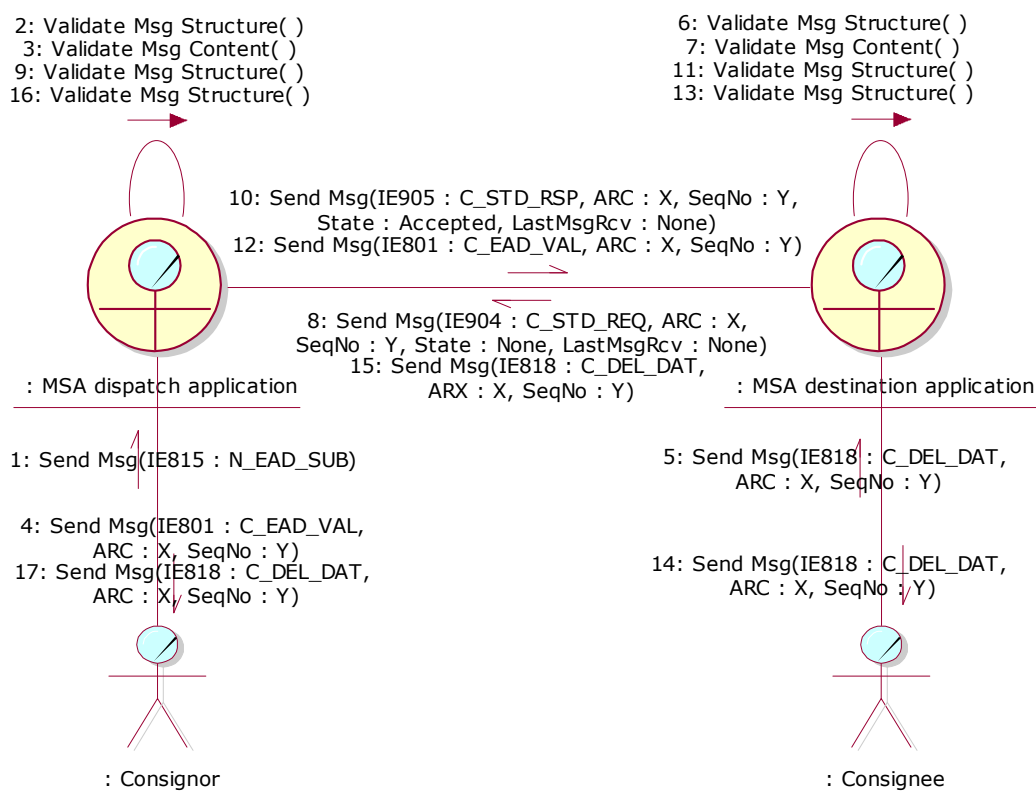


Figure 139: CLD - Positive response on a requested e-AD

III.VI.2.4.1.2 Unknown e-AD

In the case that the requested ARC is found to be unknown to the MSA dispatch application, an IE905 is generated with “Status = None” and the sequence number set to the value “1” (since the ARC is not known at MSA of Dispatch) and is sent to the MSA destination application.

When such a response is received from the MSA dispatch application, the MSA destination application rejects the draft RoR. The communication of rejection regarding the RoR is a national issue. However, it is recommended that the functional errors to be forwarded to the Consignee using the refusal message (IE704: N_REJ_DAT).

This scenario is depicted in Figure 140 and Figure 141:

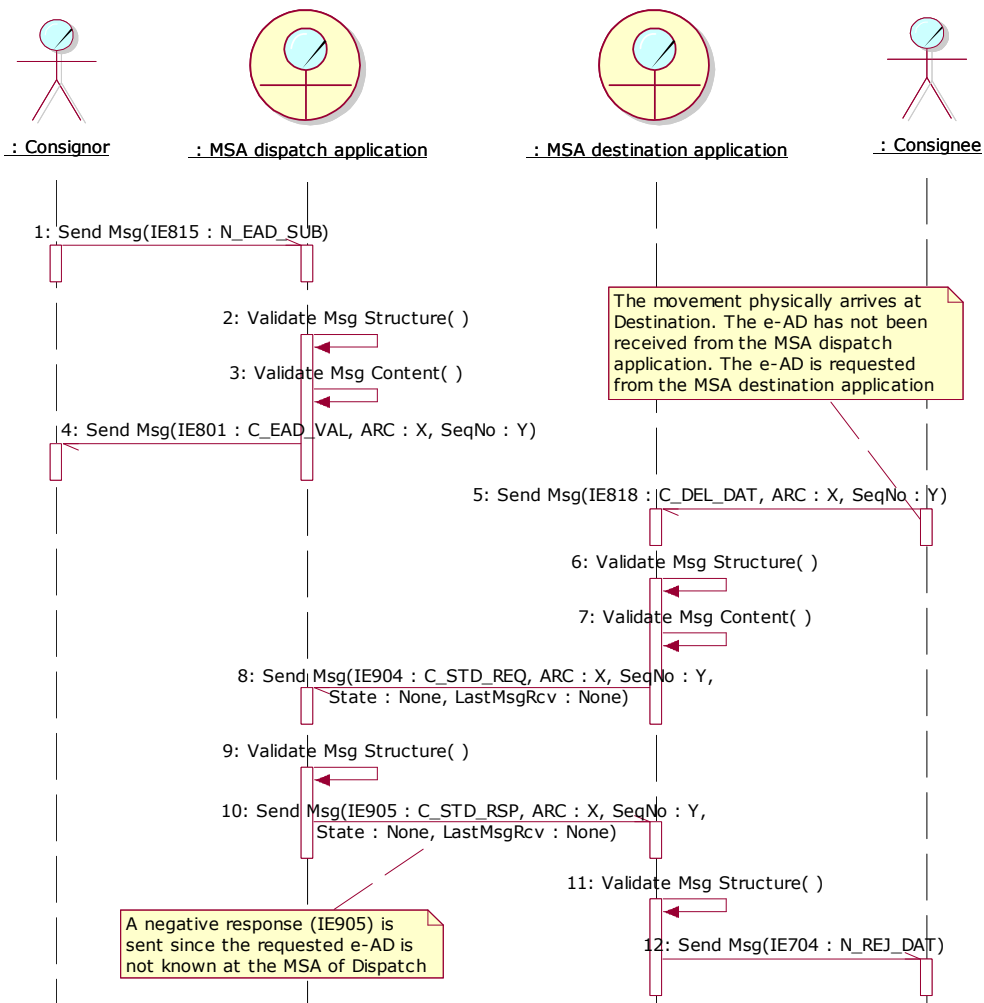


Figure 140: TSD - Negative response on a requested e-AD

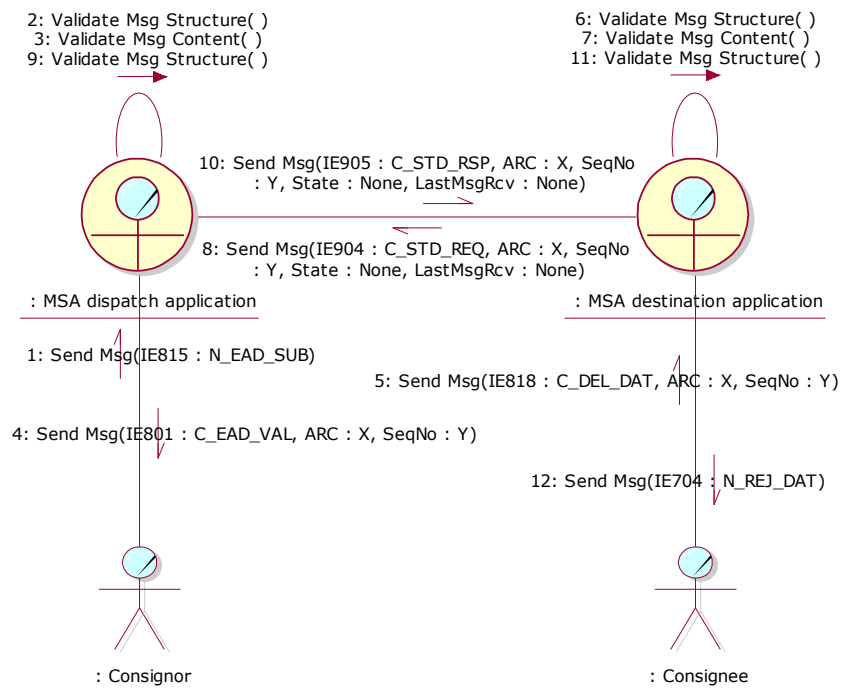


Figure 141: CLD - Negative response on a requested e-AD

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

III.VI.2.4.2 TIM_EAD timer expiration after Missed RoR

The scenario described in III.I.7.1 (Status Request/Response after the TIM_EAD timer expiration) includes the automatic triggering of Status Request/Response by the MSA dispatch application to examine why the TIM_EAD timer expired without receiving the RoR. This mechanism enables the MSA dispatch application to identify when the RoR has not been received due to technical problems and when the RoR has not been sent by the Consignee within the allocated time. The scenario of this section considers the case when the RoR has not been received due to technical problems.

An e-AD is submitted and registered successfully in both MSA dispatch and MSA destination applications (Chapter III.I.1). Moreover, the Consignee sends a RoR to the MSA destination application (Chapter III.I.4). The MSA destination application records successfully the RoR, sets the state of the movement to the “*Delivered*” or “*Partially Refused*” or “*Refused*” state and forwards it to the MSA dispatch application. For some reason, the RoR (IE818: C_DEL_DAT) is not received by the MSA dispatch application (missed/delayed RoR) until the expiration of TIM_EAD timer.

Hence, the MSA dispatch application sends a Status Request (IE904: C_STD_REQ) to the MSA destination application by including the ARC and the sequence number of the last business (event) message sent to the MSA of Destination and the information that the e-AD at MSA of Dispatch is found on the “*Accepted*” state and no message has been received from the MSA destination application. In addition, the indication of a Status Synchronisation Request in the IE904 will be automatically set by the application.

The MSA destination application receives the Status Request (IE904: C_STD_REQ) and examines the contained information. The MSA destination application realises that the MSA dispatch has not received the RoR (IE818: C_DEL_DAT) since it is found on the “*Accepted*” state and no message has been received from the MSA destination application before. Automatically, the MSA destination application:

- Sends the Status Response (IE905: C_STD_RSP) message mentioning the last known sequence number of the movement and that the e-AD on the MSA destination application is found in the “*Delivered*” or “*Partially Refused*” or “*Refused*” state and the last received message from the MSA dispatch application is the e-AD information (IE801: C_EAD_VAL);
- Regenerates the RoR (IE818: C_DEL_DAT) and sends it to the MSA dispatch application in order to update the movement state at Dispatch. The “Common Message Header” is regenerated but the “Body” is sent as initially (see Section III.VI.2.1.2).

Upon the reception of the Status Response (IE905: C_STD_RSP) from the MSA destination application, the MSA dispatch application understands that it is not the Consignee responsible for not sending the RoR, since the MSA destination application is already found on the “*Delivered*” or “*Partially Refused*” or “*Refused*” state. This means that the MSA dispatch application does not send the reminder asking for explanations but it waits for the reception of the regenerated message. When the RoR (IE818: C_DEL_DAT) is received, it is registered at the MSA of Dispatch and the state of the e-AD changes to “*Delivered*” or “*Partially Refused*” or “*Refused*”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The complete message exchange for this scenario is shown in Figure 142 and Figure 143:

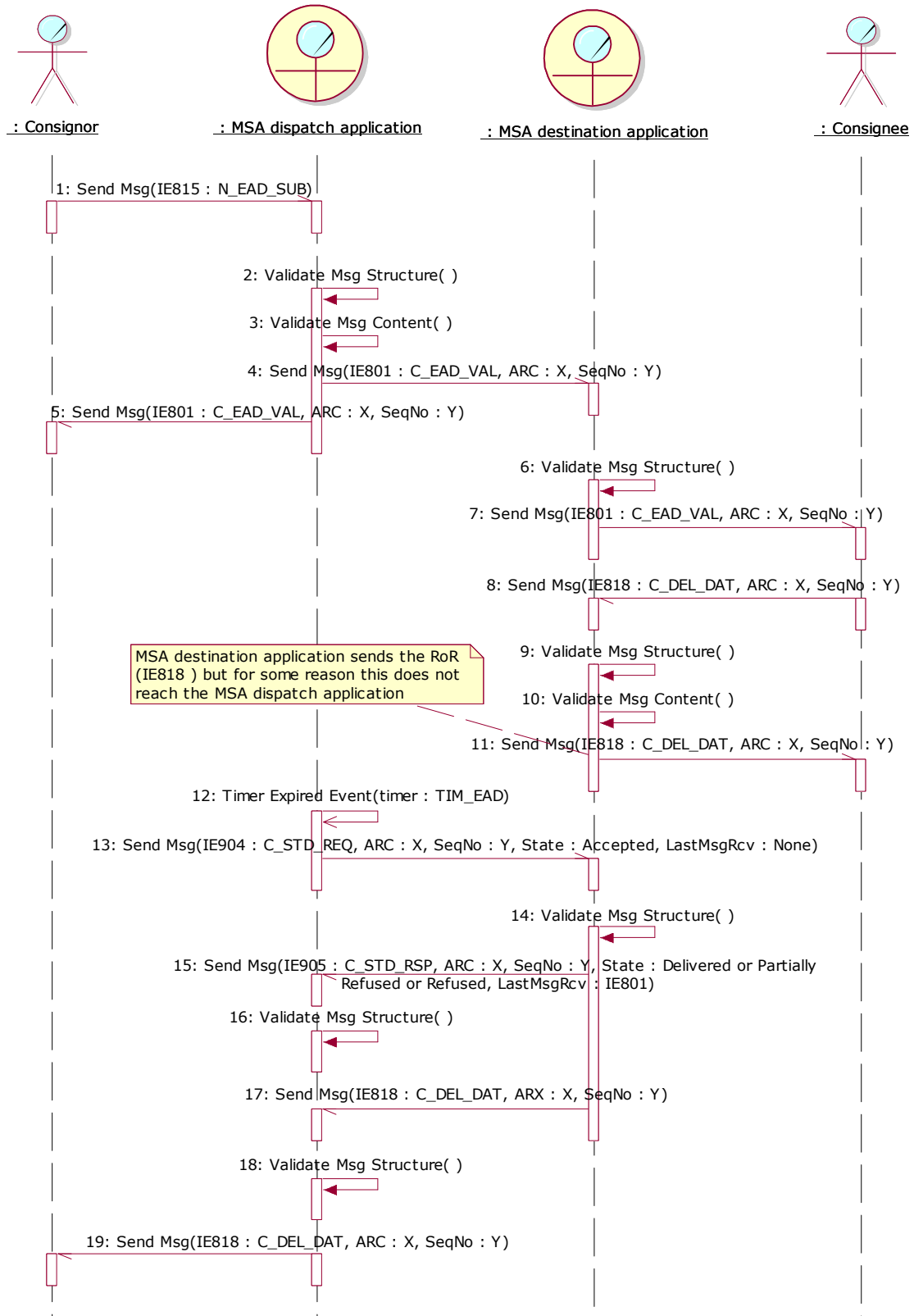


Figure 142: TSD - Status Request/Response after the TIM_EAD timer expiration - Missed RoR

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

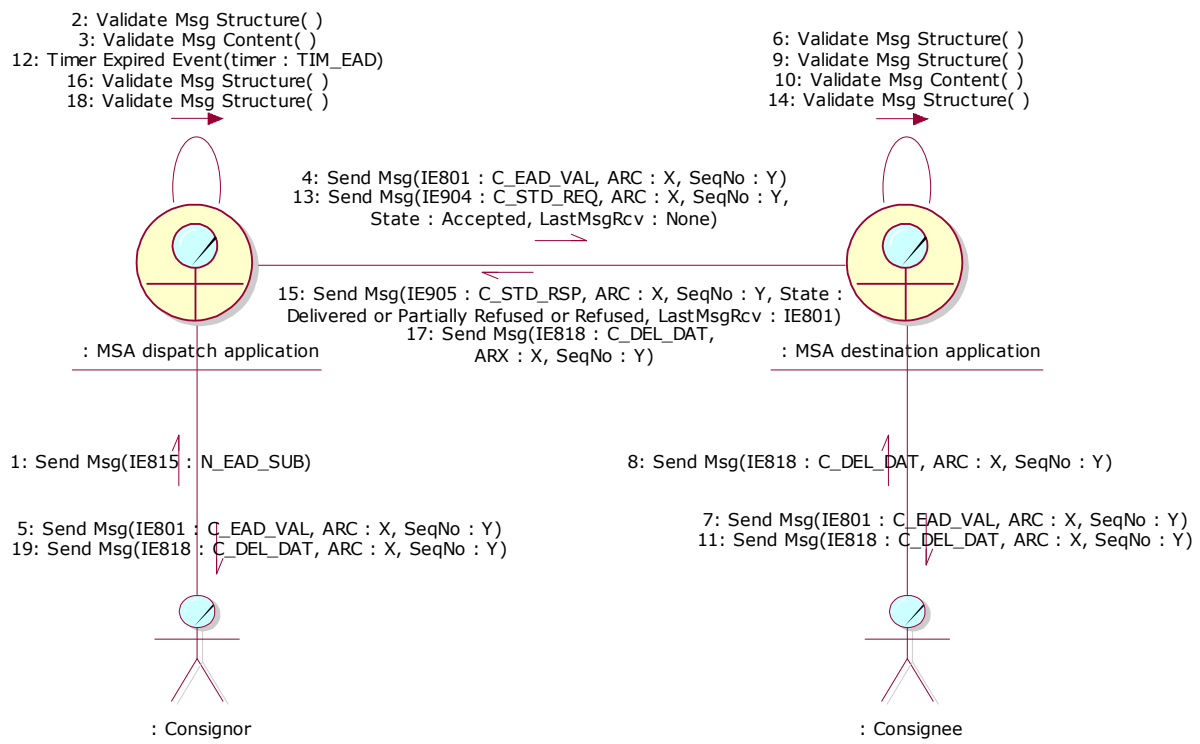


Figure 143: CLD - Status Request/Response after the TIM_EAD timer expiration - Missed RoR

III.VI.3 Manual Closing of the Movement

This scenario has been added to support the “Interoperability Issues due to the Co-existence of Paper-based System & EMCS” as this has been described in the SD [A2].

The manual closing of the movement functionality is intended to be used as an exception handling mechanism only for movements that cannot be closed in the system because the procedure to be followed is not supported electronically. An indicative example is the case of splitting "Refused" movements in FS0/FS1 (that is, movements that have been initiated electronically, have been refused electronically by the Consignee and have been split paper-based by the Consignor, as the splitting procedure is not supported electronically in FS0/FS1). However, other cases where the manual closure functionality could be needed also exist, i.e. when events and control during movements happen (as defined in FESS Section IV [A1], an event is any occurrence that is considered worth signalling to the MSAs, for instance loss, destruction or theft of a document, of part, or all of the goods, etc.); either directly following an event report (e.g. total loss of goods), or following a control, or for any other reason, such as an ascertained fraud, a MSA is entitled to interrupt the movement. Interruption of the movement is notified to all MSAs concerned by the movement, to the Consignor, to the Consignee, etc. Due to the fact that events and controls electronic management should be implemented during FS2, movements could be closed using this functionality till it is in place.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section III - Core Business	

The recommended solution is the MSA Official at Dispatch to set manually the movement as “e-AD Manually Closed” (probably using an appropriate action). The MSA dispatch application sets the e-AD to “e-AD Manually Closed” state and sends the status of the movement (IE905: C_STD_RSP) to the MSA destination application as a notification that the movement has been closed manually at Dispatch. The MSA Dispatch will also include in the IE905: C_STD_RSP message the sequence number of the last business message sent to the MSA of Destination. Since this instance of the IE905 is a one-way message, the “Correlation Identifier” Data Item of the Common Message Header Data Group should be empty, since it does not apply for requests and one way messages.

Upon the reception of the status response (IE905: C_STD_RSP), the MSA destination application validates successfully the message and sets the state of the corresponding e-AD to “e-AD Manually Closed”.

Finally, it shall be noted that the manual closure functionality does not result to the generation of the RoR at the MSA of Destination.

The following diagrams (Figure 144 and Figure 145) illustrate, as an indicative example, the scenario of closing manually a movement that has been refused by the MSA destination application:

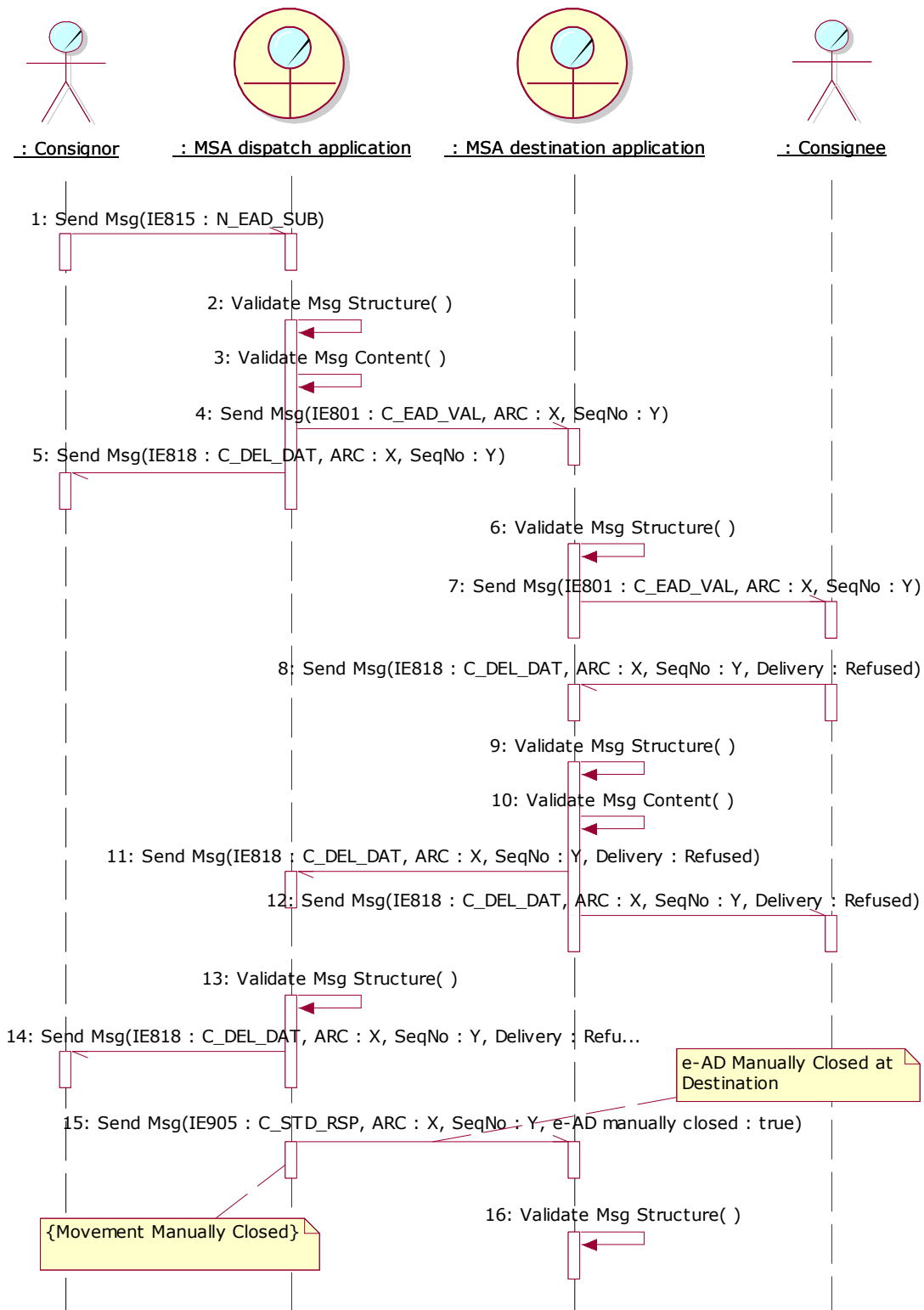


Figure 144: TSD - Manually closing of the movement after the refusal of delivery

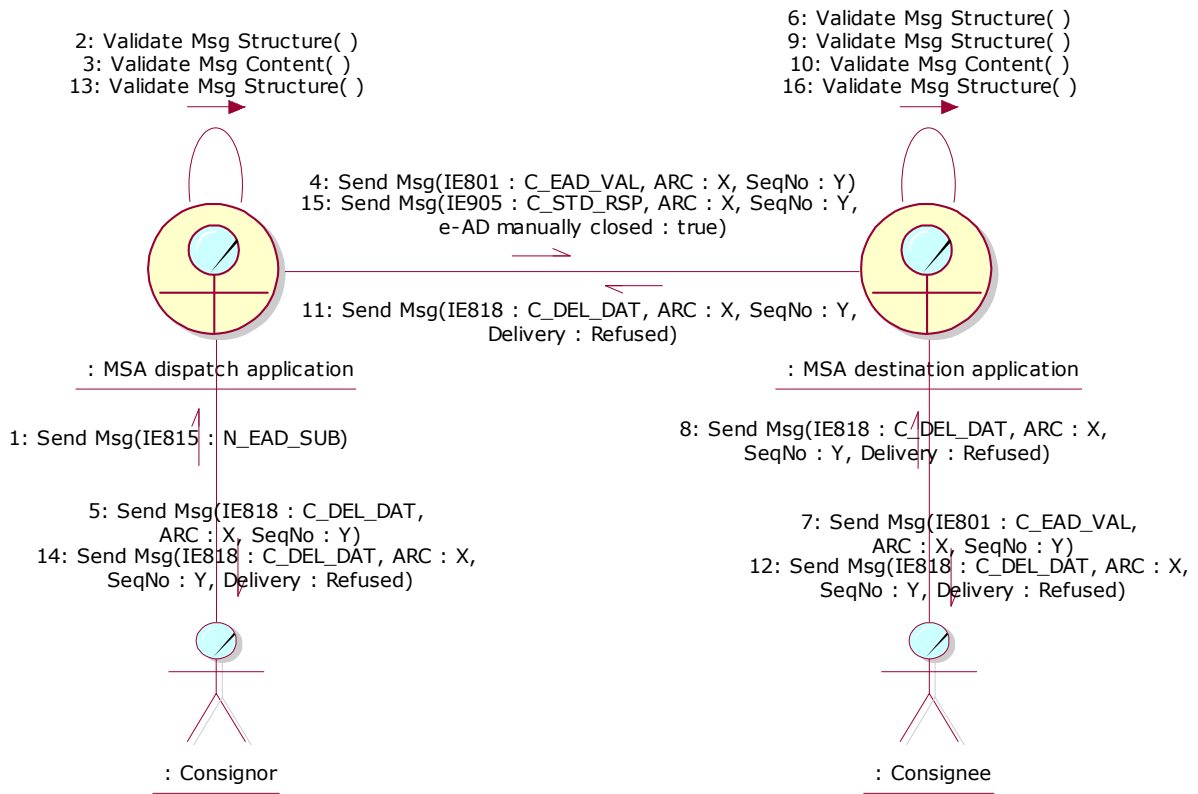


Figure 145: CLD - Manually closing of the movement after the refusal of delivery

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Section IV FOLLOW-UP AND COLLABORATION

Sub-Section IV.I Follow-up scenarios

IV.I.1 Submission of an event report (UC3.24)

Events occurring during excise movements having potential consequences on the results of the movement are reported to an Excise office. After examining the facts of an event, an Excise officer judges whether the event is worth entering into investigation procedures and/or that complementary controls must be achieved. This use case describes the message exchange protocol for submission and dissemination of an event report to the concerned MSAs.

The MSA of event in FESS [A1] is defined as the MSA where the event occurred, which also includes the MSA of Dispatch and the MSA of Destination. In the following scenarios, however, the MSA of event role is simplified and decomposed, hence wherever the MSA of event role is used it will be a third MSA (i.e. neither the MSA of Dispatch nor the MSA of Destination). The cases where the MSA of event is either the MSA of Dispatch or the MSA of Destination will be covered with the concrete roles (MSA of Dispatch or the MSA of Destination respectively) rather than the more abstract role MSA of event.

The Excise officer submitting the event report may reside at the MSA of Dispatch, the MSA of Destination, or a third MSA where the event occurred. If the Excise officer wishes to complement an existing event report, then the event report type should be set to “Complementary submission” and the event report number to be complemented should be specified. Alternatively, if the Excise officer wishes to create a new event report, then the report type should be set to “Initial submission” without specifying an event report number.

The MSA application receiving the event report from the Excise officer performs the relevant validations to the received event report (IE840: C_EVT_DAT), which pass successfully. In case the MSA application receiving the event report is different from the MSA of event, the former sends the event report to the latter. In the opposite case, the event report sending is considered non-applicable, but the MSA application receiving the event report from the Excise officer will also be the MSA of event, so the processing of the following paragraph is applicable.

The MSA where the actual event took place (MSA of event) should assign a unique reference (the structure of event report reference and the Check Digit algorithm are defined in FESS Appendix B [A1]) to the event report (in case of initial submission) or should update the existing event report (in case of complementary submission). It should then disseminate the valid (possibly complementary) event report IE840 to all concerned parties (i.e. to the MSA of Dispatch and MSA of Destination).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Whenever the MSA where the actual event took place (MSA event application) does not know all the addressees, it should perform a “Download of an e-AD” scenario (section III.I.12 Download of an e-AD (UC2.51)) in order to identify the addressees and disseminate the validated IE840 (Event Report Message Type is “Validated document”). For example, when the event took place in a third MSA (that is, it is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.1.1)) the event report (IE840:C_EVT_DAT) shall be disseminated to the MSA of Dispatch and the MSA of Destination. The MSA of Dispatch can be derived from the ARC of the movement (the third and fourth characters of the ARC denote the MSA of Dispatch). However, the current MSA of Destination will not be known. Therefore, the MSA of Destination will not become known unless a “Download of an e-AD” scenario is executed (see III.I.12 Download of an e-AD (UC2.51)).

The scenarios of this section describe the “Submission of an event report (UC3.24)” use case, through the following scenarios:

- an event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.1.1);
- an event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch (see IV.I.1.2);
- an event report submitted by the MSA of Destination and event occurred at the MSA of Destination (see IV.I.1.3);
- an event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination (see IV.I.1.4);
- an event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination (see IV.I.1.5);
- an event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch (see IV.I.1.6);
- an event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination (see IV.I.1.7).

IV.I.1.1 Event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination

This scenario assumes that the submitting Excise officer resides at the MSA where the event occurred that is neither the MSA of Dispatch nor the MSA of Destination (i.e. a third MSA). The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA event application. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) directly to the MSA where the event took place, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA event application is not applicable.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Since the MSA of event has not been involved in the movement (or has been involved in the movement as a former MSA of Destination), it has not received any IE messages concerning the movement (or has received some IE messages that do not reflect the current state of the movement). Thus, the MSA destination application is not known to the MSA event application for disseminating the validated document (Event Report Message Type is “Validated submission”).

The MSA event application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that in case that the MSA event application has not received any message that is related to the concerned ARC, then it has to set the “Status” Data Item to “None” and the “Last Received Message Type” Data Item to “None”, since the latter Data Item reflects the last message received from the MSA dispatch application. Furthermore, in case that the MSA event application is a former MSA of Destination with respect to the specific ARC, then it has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA event application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA event application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA event application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA event application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA event application has all the information for the concerned movement via the IE934 message. The MSA event application identifies the MSA of Destination and disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA dispatch application and to the MSA destination application.

The MSA dispatch application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignor.

The MSA destination application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA event application transfers the event report (IE840: C_EVT_DAT) to the Excise officer that submitted the event report originally, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The scenario for an event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 146 and Figure 147:

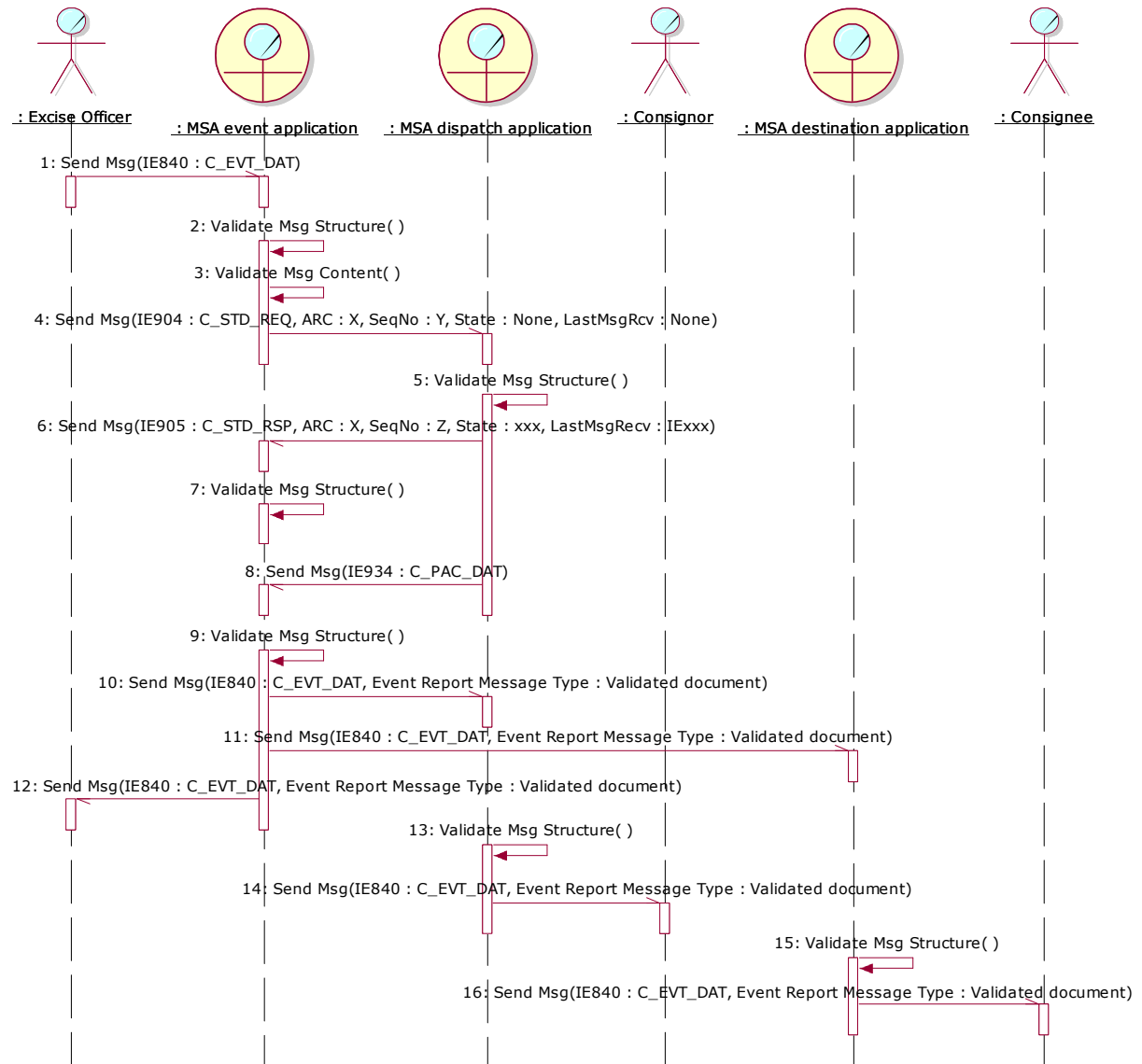


Figure 146: TSD - Event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination

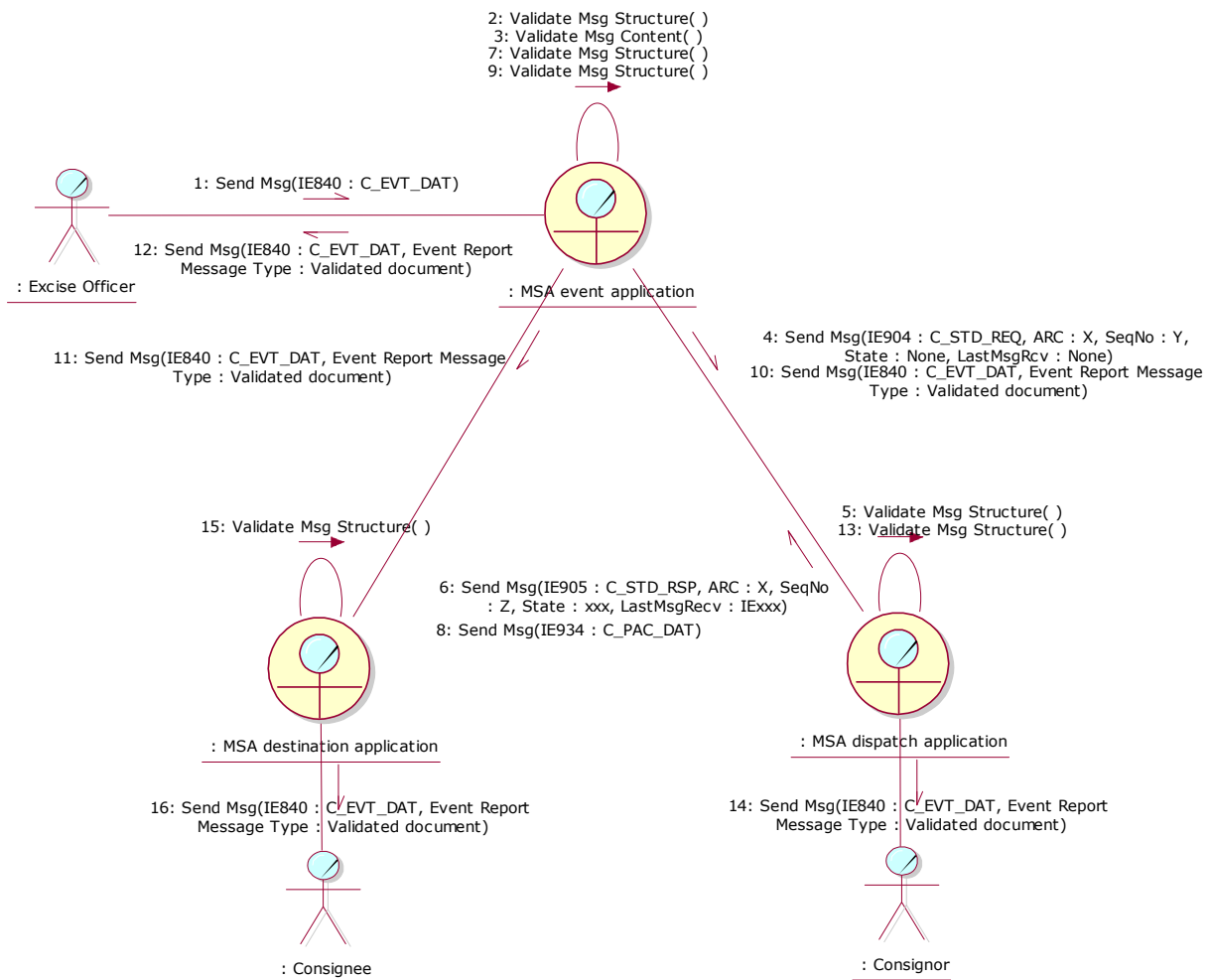


Figure 147: CLD - Event report submitted by the MSA of Event that is neither the MSA of Dispatch nor the MSA of Destination

IV.I.1.2 Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch

This scenario assumes that the submitting Excise officer resides at the MSA where the event occurred; that is the MSA of Dispatch. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA event application. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) directly to the MSA where the event took place, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA event application is not applicable.

Since the MSA of Dispatch is also the MSA of event, thus the MSA destination application is known for disseminating the validated document (Event Report Message Type is “Validated submission”).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA dispatch application disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA destination application, to the Consignor, and to the Excise officer that submitted the event report originally, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The MSA destination application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee.

The scenario of an event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch is depicted in Figure 148 and Figure 149:

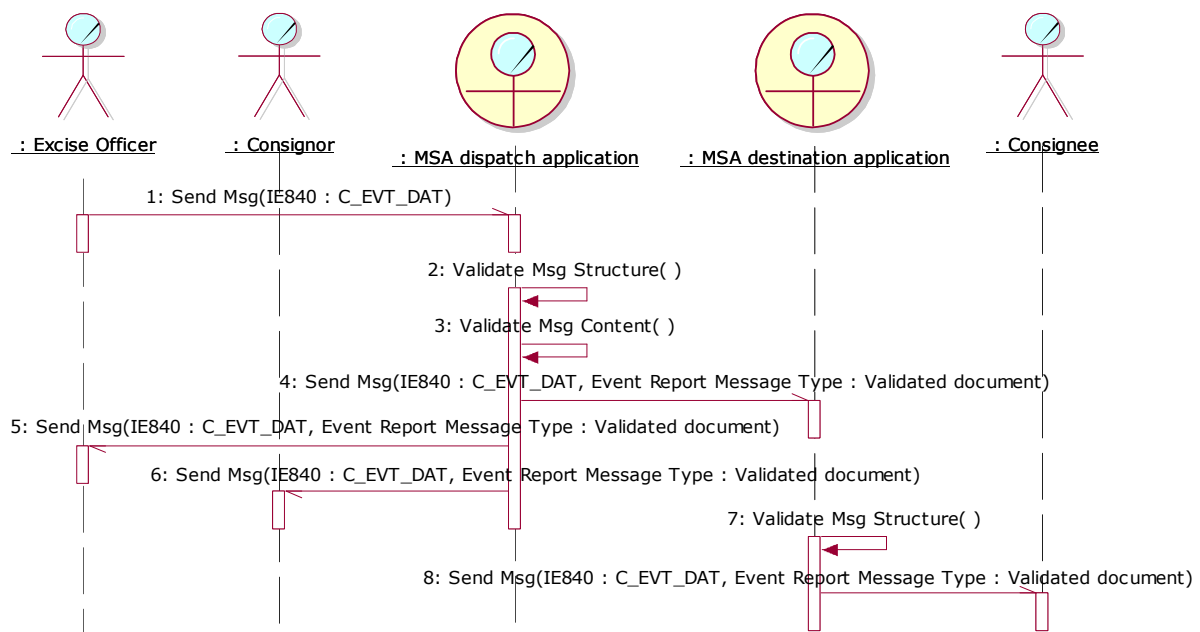


Figure 148: TSD – Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

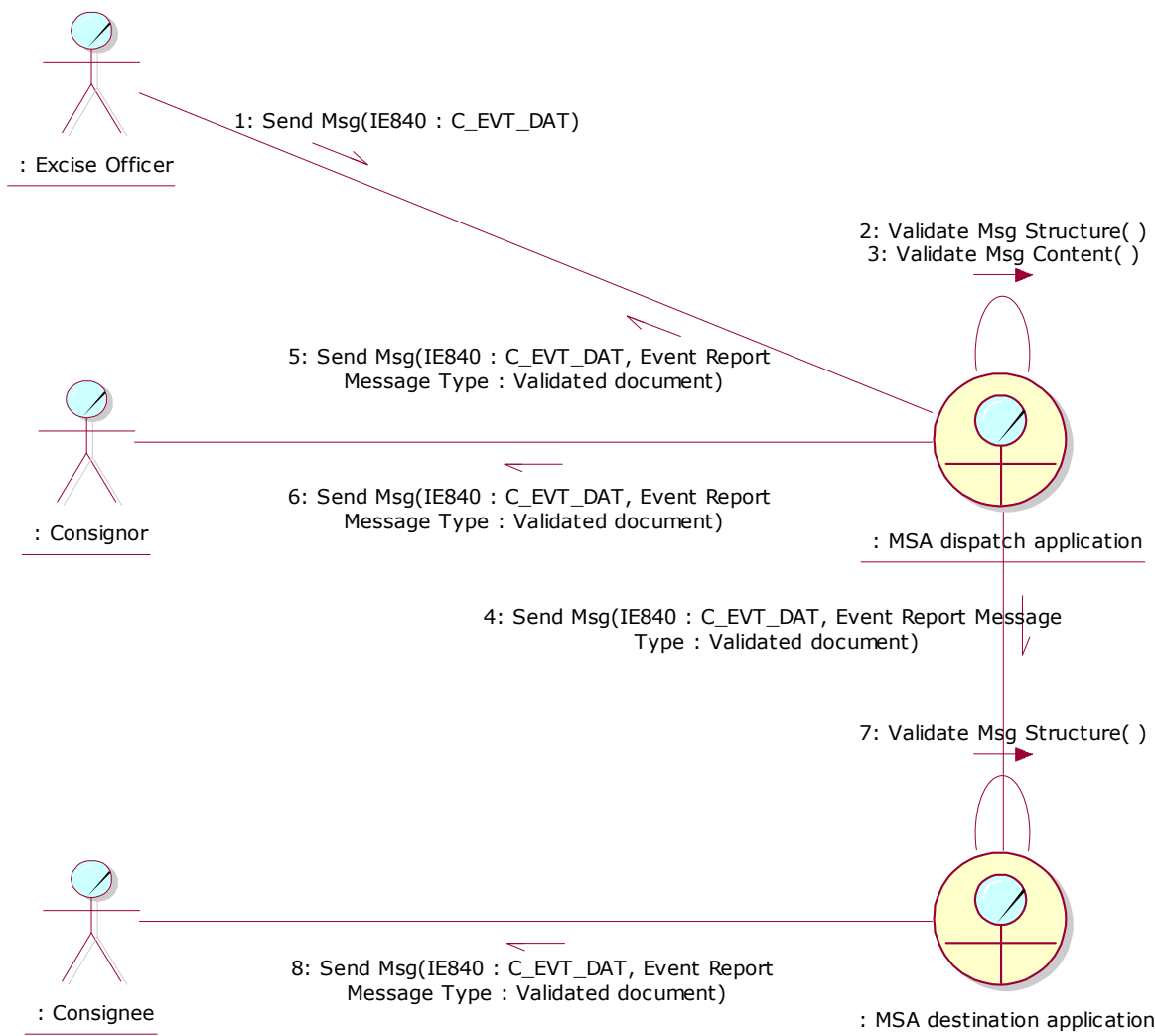


Figure 149: CLD – Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Dispatch

IV.I.1.3 Event report submitted by the MSA of Destination and event occurred at the MSA of Destination

This scenario assumes that the submitting Excise officer resides at the MSA where the event occurred; that is the MSA of Destination. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA event application. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) directly to the MSA where the event took place, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA event application is not applicable.

The MSA dispatch application is known to the MSA destination application, which is also the MSA of event, for disseminating the validated document (Event Report Message Type is “Validated submission”). The MSA of Destination should determine that it is synchronised with the MSA of Dispatch, thus confirming it is the current MSA of Destination rather than a former MSA of Destination.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA destination application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that the MSA destination application has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA destination application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA destination application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA destination application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA destination application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA destination application has all the information for the concerned movement via the IE934 message. It is indeed identified that the MSA destination application is the current MSA of Destination. Thus, the MSA destination application disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA dispatch application, to the Consignee, and to the Excise officer that submitted the event report originally, hence confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The MSA dispatch application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee.

The scenario for an event report submitted by the MSA of Destination and event occurred at the MSA of Destination is depicted in Figure 150 and Figure 151:

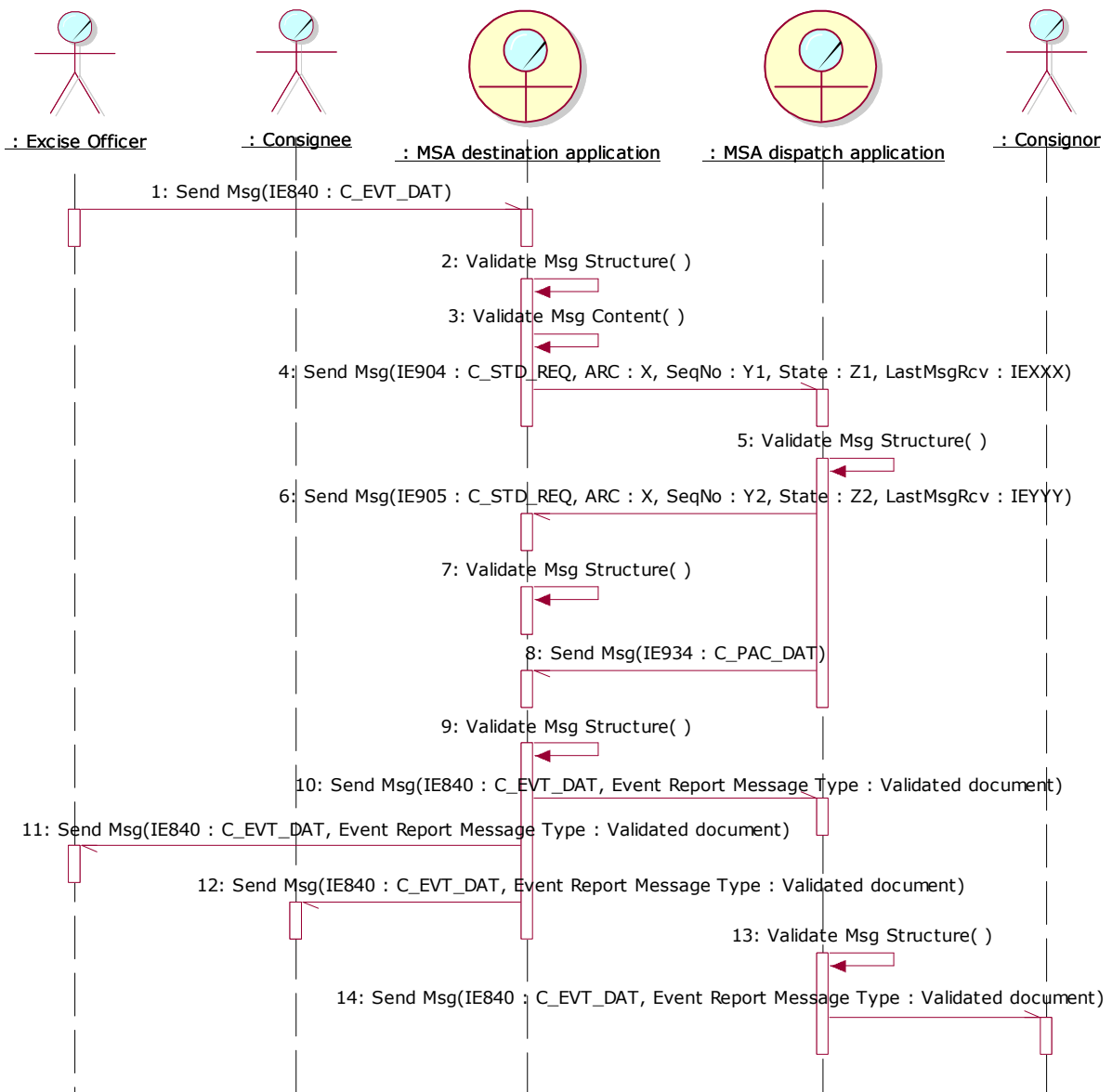


Figure 150: TSD - Event report submitted by the MSA of Destination and event occurred at the MSA of Destination

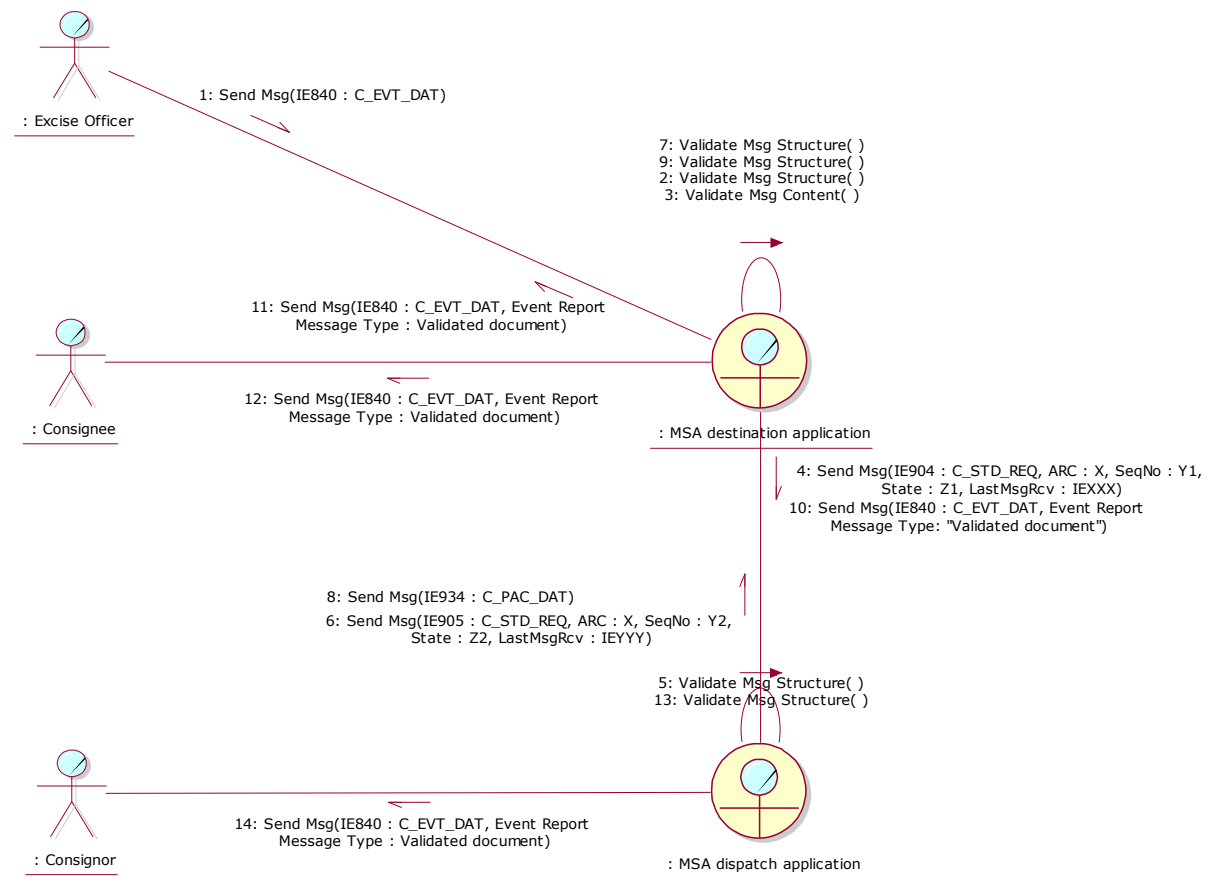


Figure 151: CLD - Event report submitted by the MSA of Destination and event occurred at the MSA of Destination

IV.I.1.4 Event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

This scenario assumes that the submitting Excise officer resides at the MSA of Dispatch, which is not the MSA of event. The event occurred at neither the MSA of Dispatch nor the MSA of Destination. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA event application. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) to the MSA dispatch application and not directly to the MSA of event, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA event application is applicable. The MSA dispatch application initially submits the event report (IE840: C_EVT_DAT) to the MSA event application (Event Report Message Type is “Initial submission”).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA event application receives and validates the event report (IE840: C_EVT_DAT) successfully. Since the MSA of event has not been involved in the movement (or has been involved in the movement as a former MSA of Destination), it has not received any IE messages concerning the movement (or has received some IE messages that do not reflect the current state of the movement). Thus, the MSA destination application is not known to the MSA event application for disseminating the validated document (Event Report Message Type is “Validated submission”). The MSA event application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that in case that the MSA event application has not received any message that is related to the concerned ARC, then it has to set the “Status” Data Item to “None” and the “Last Received Message Type” Data Item to “None”, since the latter Data Item reflects the last message received from the MSA dispatch application. Furthermore, in case that the MSA event application is a former MSA of Destination with respect to the specific ARC, then it has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA event application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA event application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA event application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA event application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA event application has all the information for the concerned movement via the IE934 message. The MSA event application identifies the MSA of Destination and disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA dispatch application and to the MSA destination application.

The MSA dispatch application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignor and to the Excise officer, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The MSA destination application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee.

The scenario for an event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 152 and Figure 153:

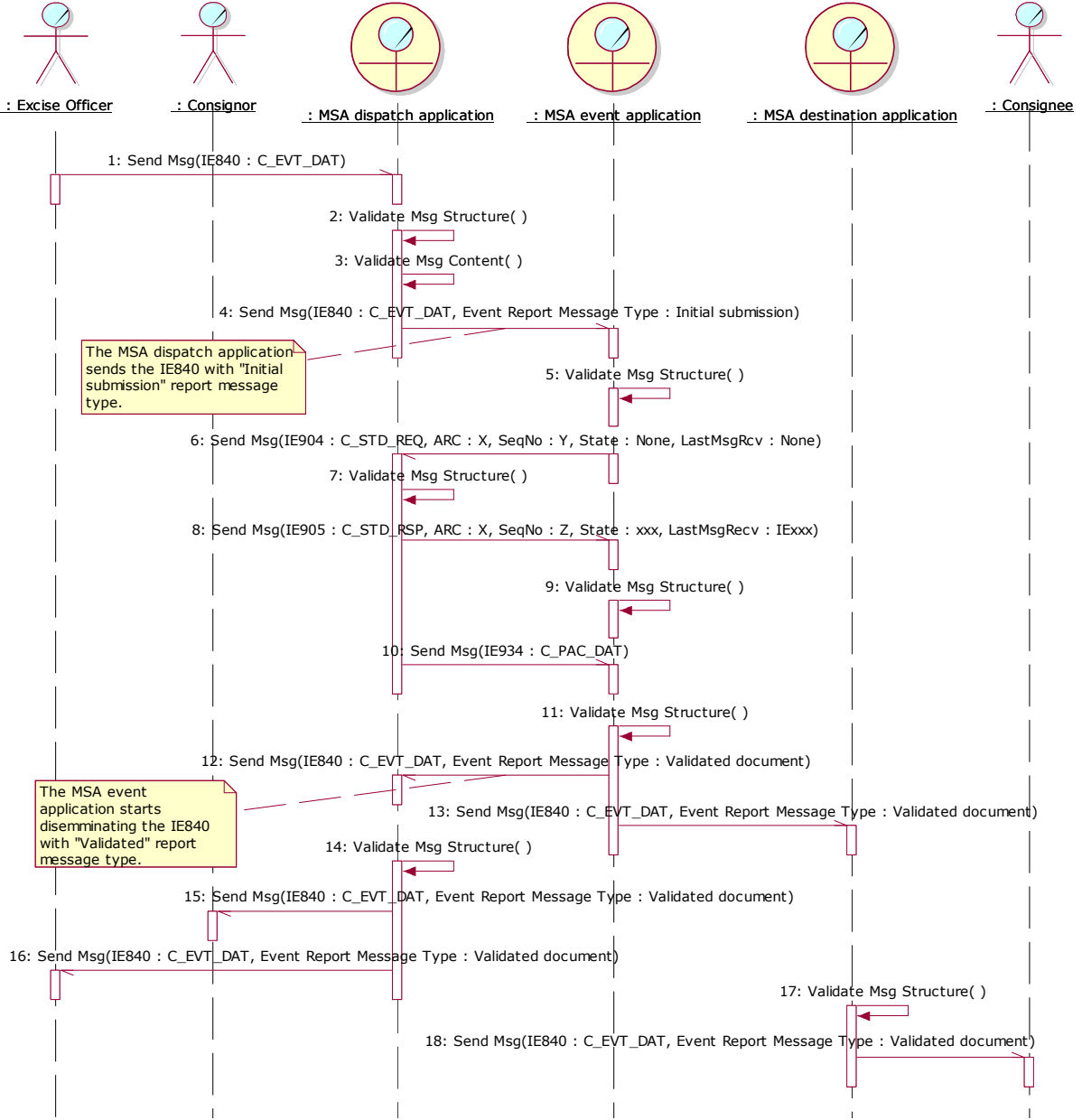


Figure 152: TSD - Event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

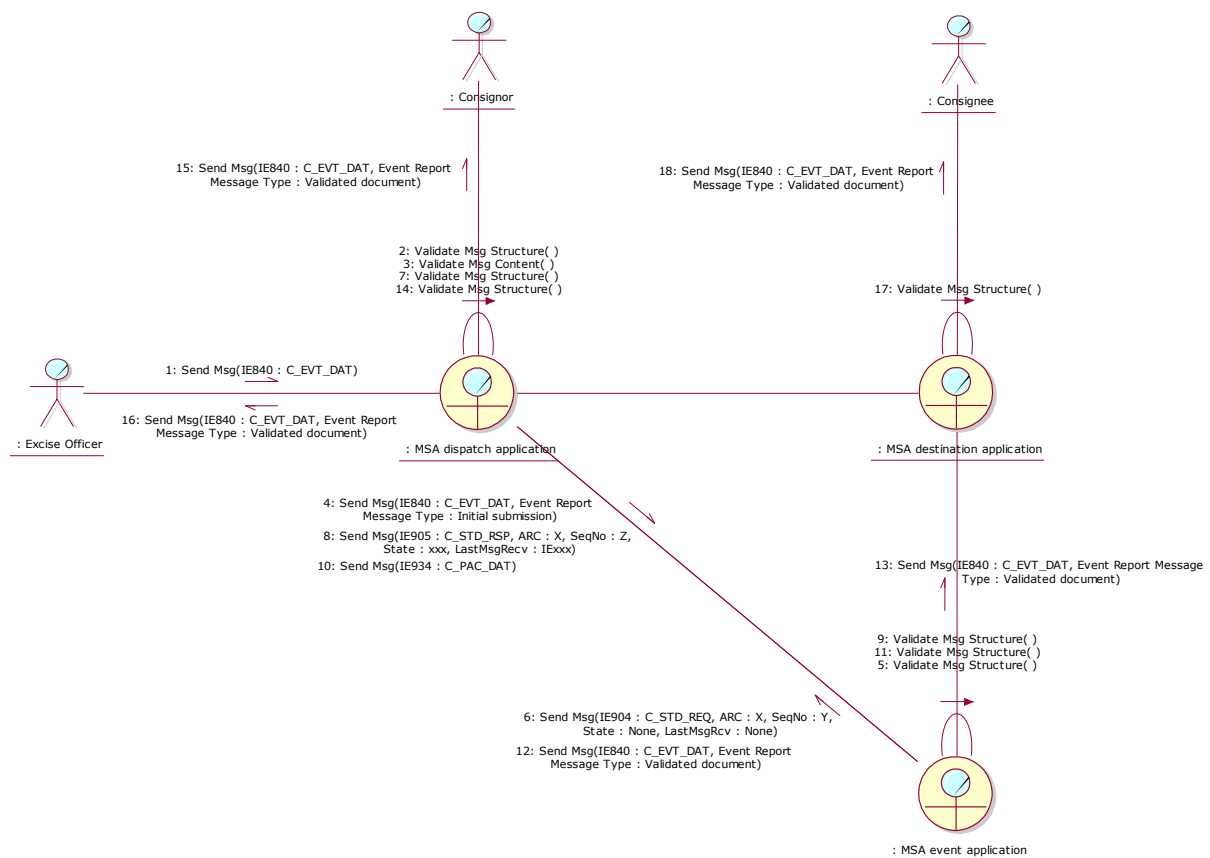


Figure 153: CLD - Event report submitted by the MSA of Dispatch and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

IV.I.1.5 Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination

This scenario assumes that the submitting Excise officer resides at the MSA of Dispatch, which is not the MSA of event. The event occurred at the MSA of Destination. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA of Destination. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) to the MSA dispatch application and not directly to the MSA destination application, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA where the event took place is applicable. The MSA dispatch application initially submits the event report (IE840: C_EVT_DAT) to the MSA destination application (Event Report Message Type is “Initial submission”).

The MSA destination application receives and validates the event report (IE840: C_EVT_DAT) successfully. The MSA of Destination should determine that it is synchronised with the MSA of Dispatch, thus confirming it is the current MSA of Destination rather than a former MSA of Destination for disseminating the validated document (Event Report Message Type is “Validated submission”).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA destination application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that the MSA destination application has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA destination application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA destination application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA destination application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA destination application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA destination application has all the information for the concerned movement via the IE934 message. It is indeed identified that the MSA destination application is the current MSA of Destination. Thus, the MSA destination application disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA dispatch application and to the Consignee.

The MSA dispatch application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignor and to the Excise officer, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The scenario for an event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination is depicted in Figure 154 and Figure 155:

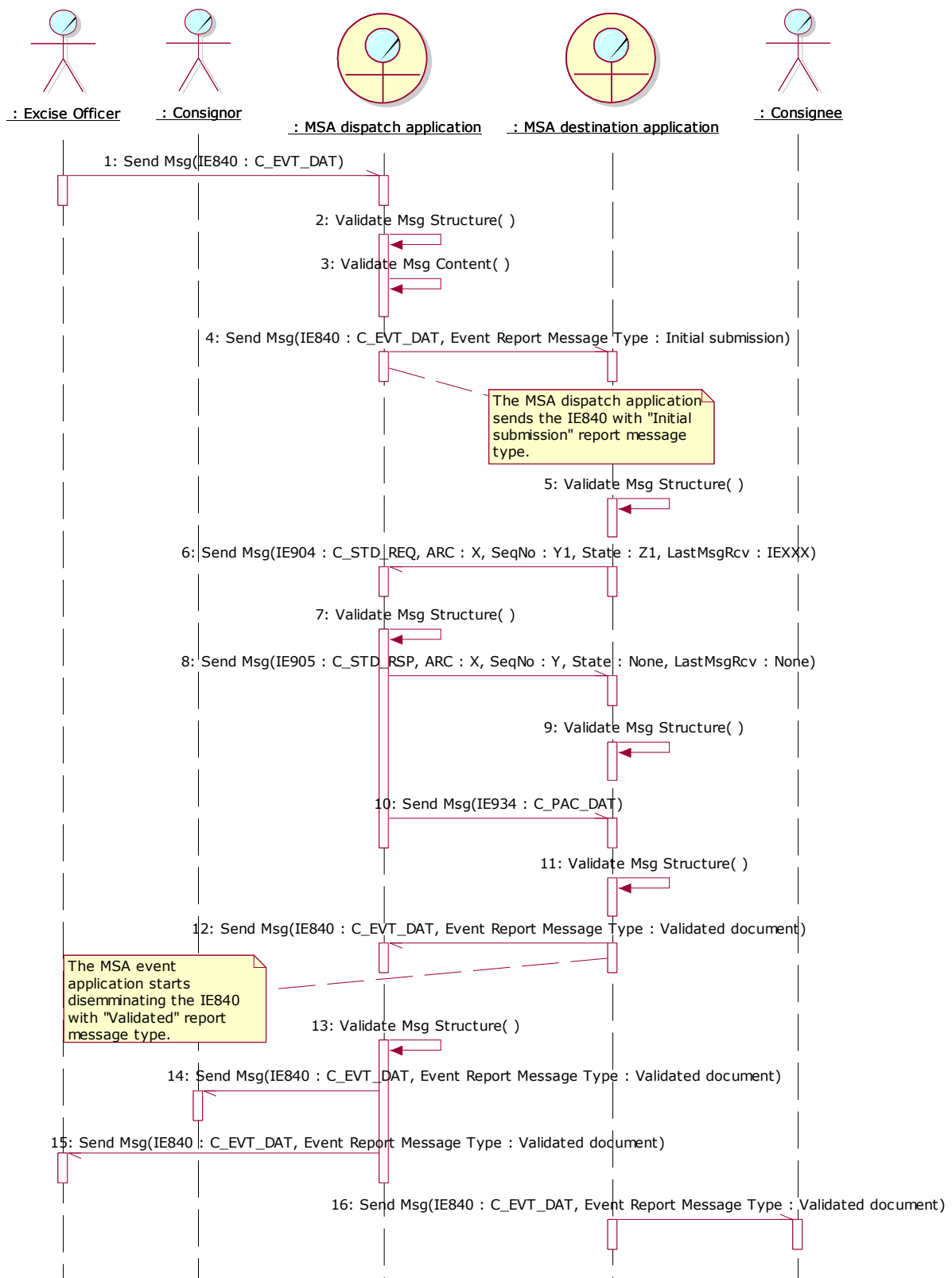


Figure 154: TSD - Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination

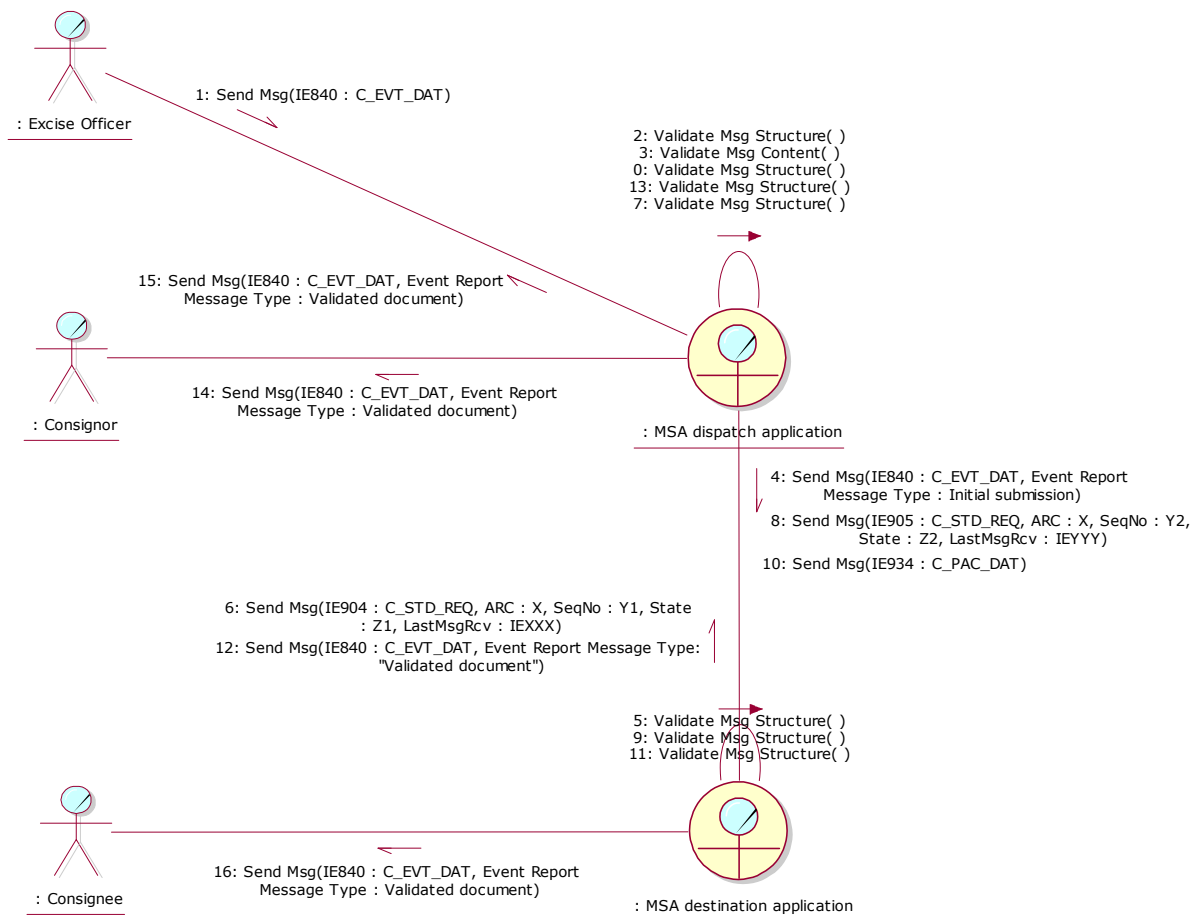


Figure 155: CLD - Event report submitted by the MSA of Dispatch and the event occurred at the MSA of Destination

IV.I.1.6 Event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch

This scenario assumes that the submitting Excise officer resides at the MSA of Destination, which is not the MSA of event. The event occurred at the MSA of Dispatch. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA of Dispatch. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) to the MSA destination application and not directly to the MSA dispatch application, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA where the event took place is applicable. The MSA destination application initially submits the event report (IE840: C_EVT_DAT) to the MSA dispatch application (Event Report Message Type is “Initial submission”).

The MSA dispatch application receives and validates the event report (IE840: C_EVT_DAT) successfully. Since the MSA of Dispatch is also the MSA of event, thus the MSA destination application is known for disseminating the validated document (Event Report Message Type is “Validated submission”). The MSA dispatch application disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA destination application and to the Consignor.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA destination application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee and to the Excise officer, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The scenario for an event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch is depicted in Figure 156 and Figure 157:

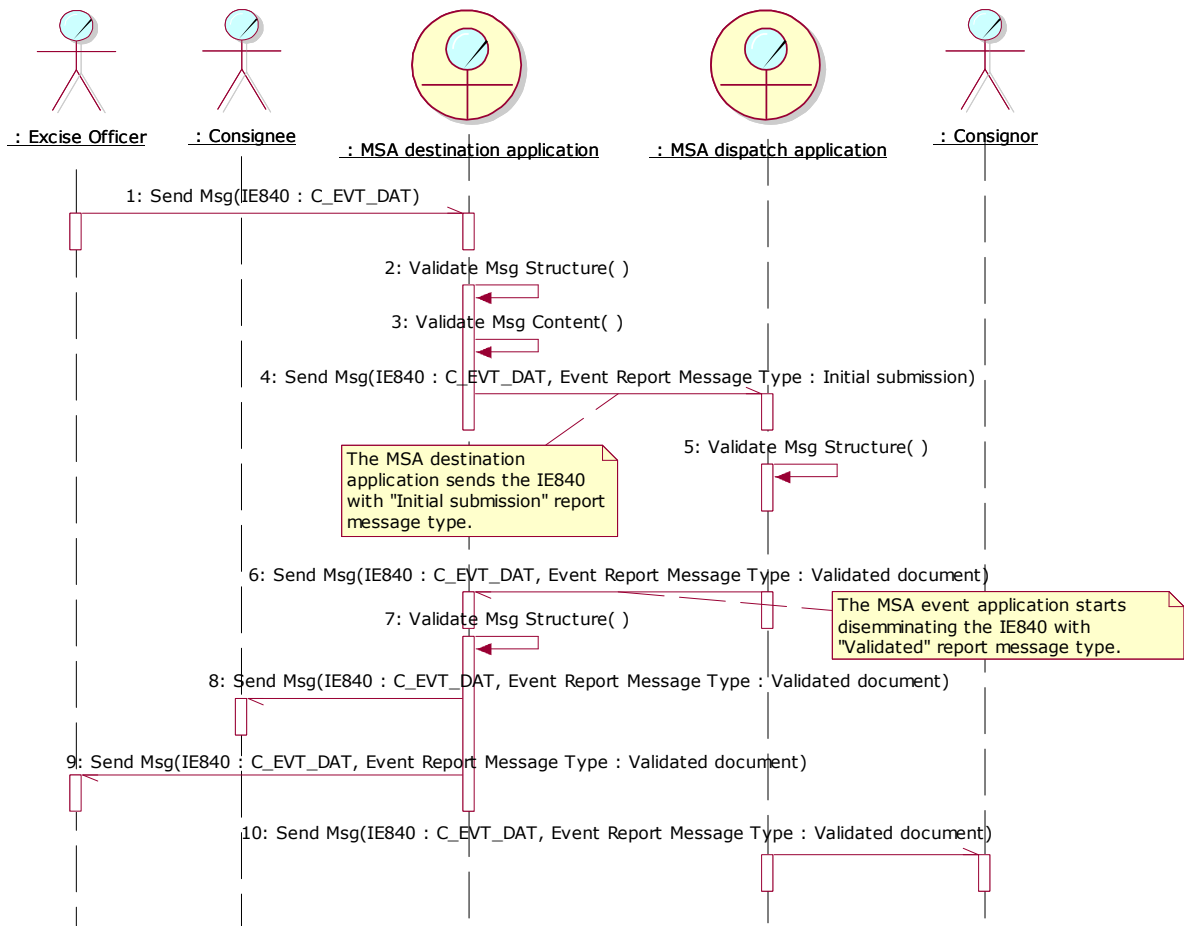


Figure 156: TSD - Event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch

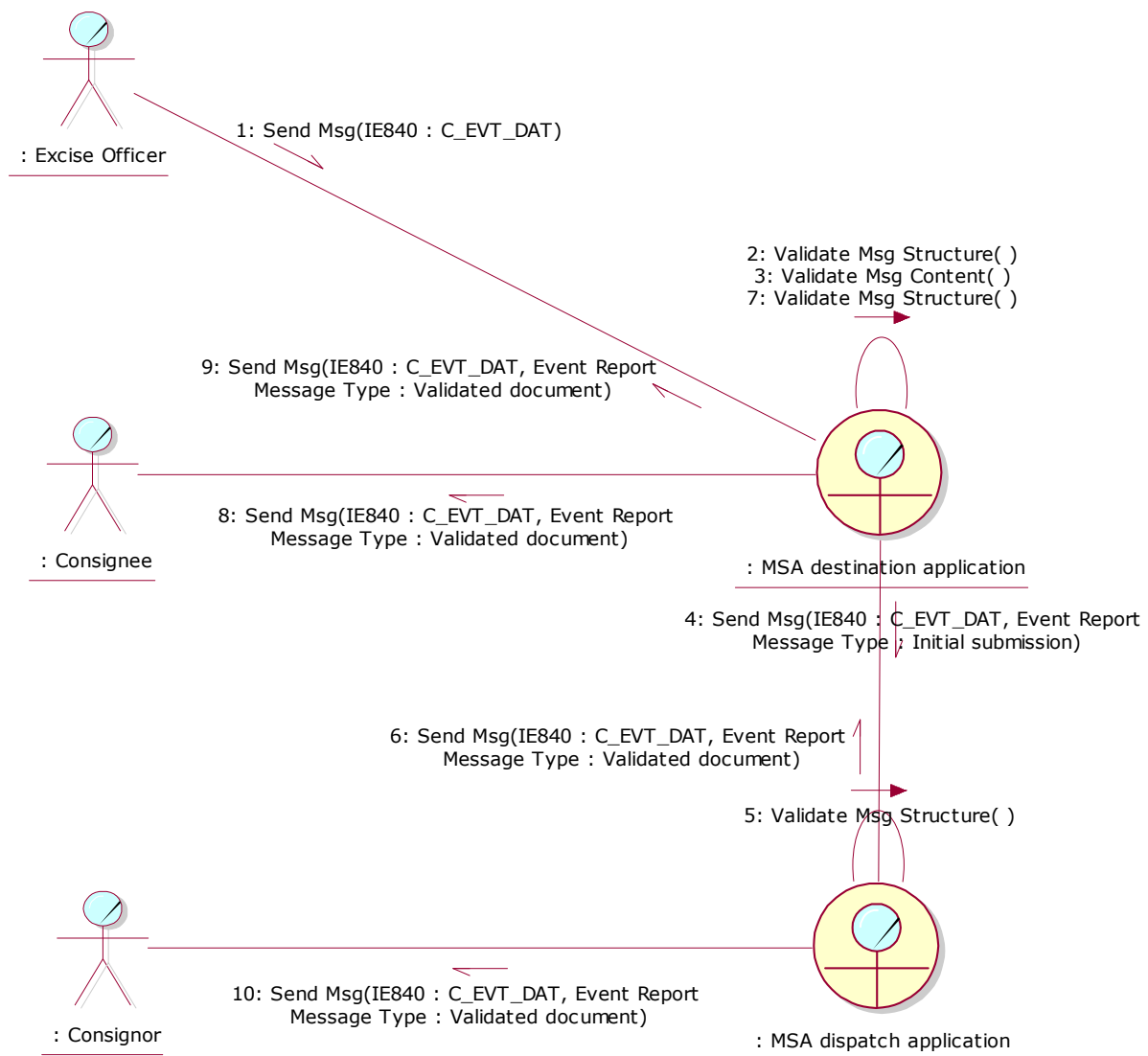


Figure 157: CLD - Event report submitted by the MSA of Destination and the event occurred at the MSA of Dispatch

IV.I.1.7 Event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

This scenario assumes that the submitting Excise officer resides at the MSA of Destination, which is not the MSA of event. The event occurred at neither the MSA of Dispatch nor the MSA of Destination. The draft event report is submitted without specifying an event report number.

The application receiving the Excise officer’s event report is considered the MSA that will submit the event report to the MSA event application. Since in the specific scenario the Excise officer submits the draft event report (IE840: C_EVT_DAT) to the MSA destination application and not directly to the MSA of event, the step of transferring the initial submission of the event report (IE840: C_EVT_DAT) to the MSA event application is applicable. The MSA destination application initially submits the event report (IE840: C_EVT_DAT) to the MSA event application (Event Report Message Type is “Initial submission”).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA event application receives and validates the event report (IE840: C_EVT_DAT) successfully. Since the MSA of event has not been involved in the movement (or has been involved in the movement as a former MSA of Destination), it has not received any IE messages concerning the movement (or has received some IE messages that do not reflect the current state of the movement). Thus, the MSA destination application is not known to the MSA event application for disseminating the validated document (Event Report Message Type is “Validated submission”). The MSA event application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that in case that the MSA event application has not received any message that is related to the concerned ARC, then it has to set the “Status” Data Item to “None” and the “Last Received Message Type” Data Item to “None”, since the latter Data Item reflects the last message received from the MSA dispatch application. Furthermore, in case that the MSA event application is a former MSA of Destination with respect to the specific ARC, then it has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA event application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA event application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA event application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA event application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA event application has all the information for the concerned movement via the IE934 message. The MSA event application identifies the MSA of Destination and disseminates the validated event report document IE840 (event report type is “Validated submission”) to the MSA dispatch application and to the MSA destination application.

The MSA dispatch application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignor.

The MSA destination application, after validating the received message structure, transfers the event report (IE840: C_EVT_DAT) to the Consignee and to the Excise officer, thus confirming that the event report has been successfully registered and disseminated to all concerned MSAs.

The scenario for an event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 158 and Figure 159:

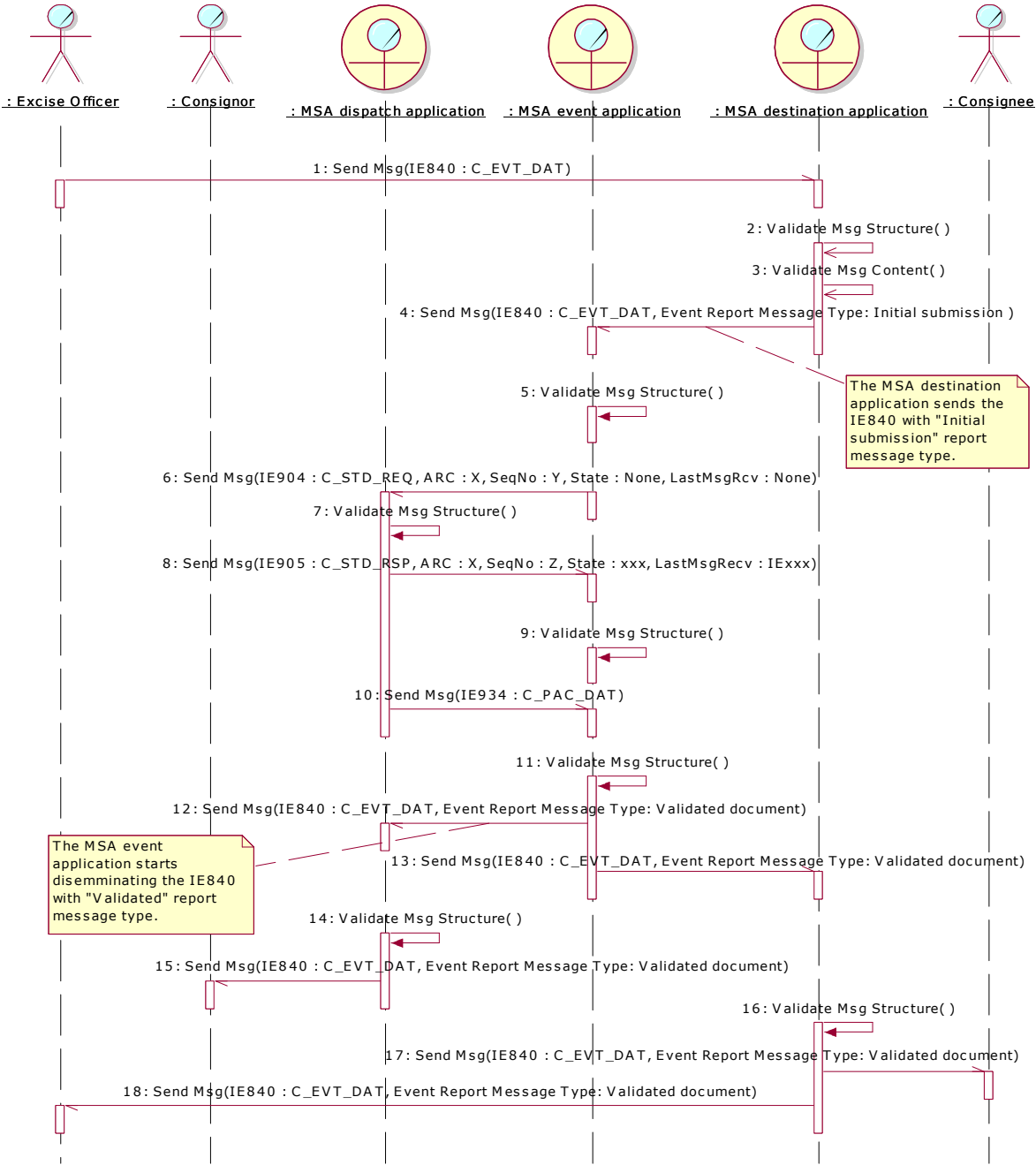


Figure 158: TSD - Event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

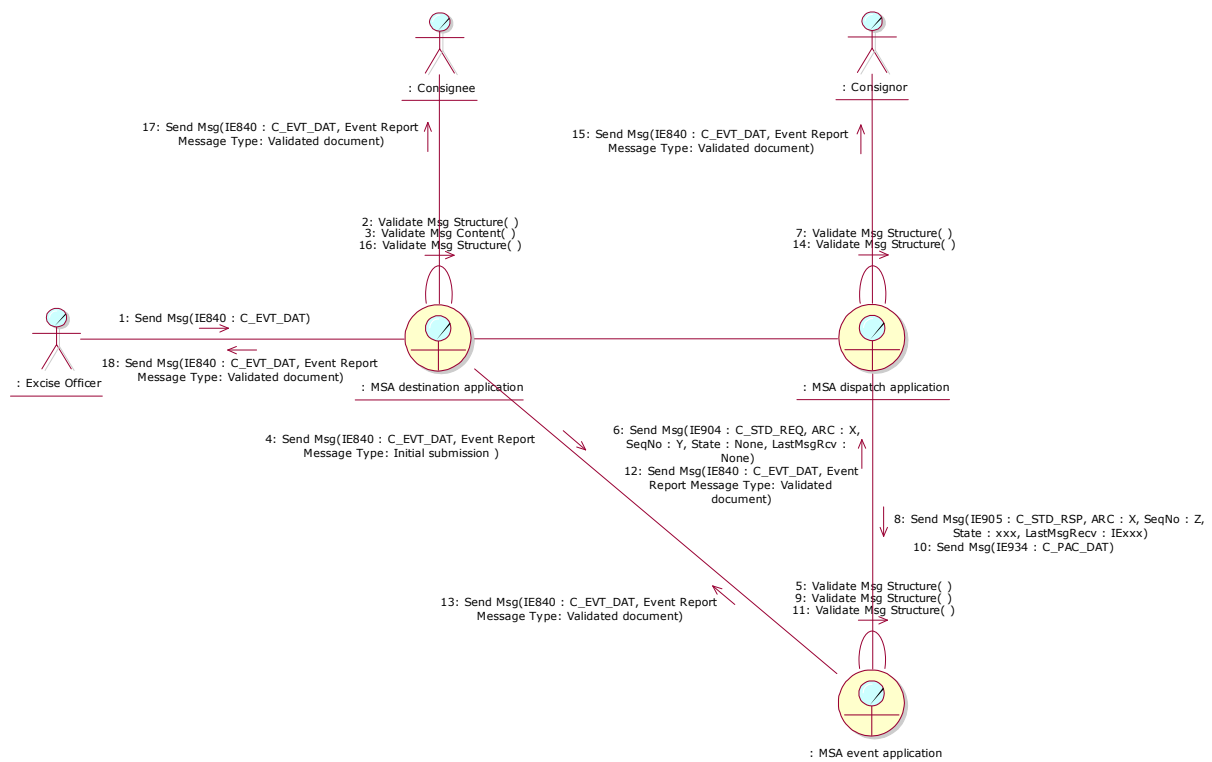


Figure 159: CLD - Event report submitted by the MSA of Destination and the event occurred at neither the MSA of Dispatch nor the MSA of Destination

IV.I.2 Control and submission of the control report (UC3.03)

An excise movement may be verified by a Control officer. During the control, the documentary information related to the e-AD and/or physical state of goods may be verified. The findings of the control are then accumulated by the Control officer into a control report. This use case describes the message exchange protocol for submission and dissemination of a control report to the concerned MSAs.

The MSA of control in FESS [A1] is defined as the MSA where the control occurred, which also includes the MSA of Dispatch and the MSA of Destination. In the following scenarios, however, the MSA of control role is simplified and decomposed, hence wherever the MSA of control is a third MSA (i.e. neither the MSA of Dispatch nor the MSA of Destination) the role third MSA will be utilised. The cases where the MSA of control is either the MSA of Dispatch or the MSA of Destination will be covered with the concrete roles (MSA of Dispatch or the MSA of Destination respectively) rather than the more abstract role MSA of control.

The MSA where the actual control took place should assign a unique reference (the structure of control report reference and the Check Digit algorithm are defined in FESS Appendix B [A1]) to the control report (in case of first submission) or should update the existing event report (in case of complementary submission). It should then disseminate the valid (possibly complementary) event report IE717 to all concerned parties (i.e. to the MSA of Dispatch and MSA of Destination).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Whenever the MSA where the actual control took place does not know all the addressees, it should perform a “Download of an e-AD” scenario (section III.I.12 Download of an e-AD (UC2.51)) in order to identify the addressees and disseminate the validated IE717 (Control Report Message Type is “Validated document”). For example, when the control took place in a third MSA (that is, it is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.2.3)) the control report (IE717: C_CCR_DAT) shall be disseminated to the MSA of Dispatch and the MSA of Destination. The MSA of Dispatch can be derived from the ARC of the movement (the third and fourth characters of the ARC denote the MSA of Dispatch). However, the current MSA of Destination will not be known. Therefore, the MSA of Destination will not become known unless a “Download of an e-AD” scenario is executed (see III.I.12 Download of an e-AD (UC2.51)).

The scenarios of this section describe the “Control and submission of the control report (UC3.03)” use case, through the following scenarios:

- a control report submitted by the MSA dispatch application (see IV.I.2.1);
- a control report submitted by the MSA destination application (see IV.I.2.2);
- a control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.2.3).

IV.I.2.1 Control report submitted by the MSA of Dispatch

This scenario assumes that the submitting Control officer resides at the MSA of Dispatch where the control occurred.

The Control officer submits his findings via a control report (IE717: C_CCR_DAT) to the MSA dispatch application, for validation and notification of the concerned MSAs. The draft control report is submitted without specifying a control report number.

The MSA dispatch application performs the relevant validations to the received control report (IE717: C_CCR_DAT), which pass successfully. If the MSA dispatch application has not received another control report prior to this reception, it assigns a unique reference number to the control report reference (the structure of control report reference and the Check Digit algorithm are defined in FESS Appendix B [A1]), otherwise it complements the existing report. Since the MSA where the control occurred is the MSA of Dispatch, the MSA destination application is known for disseminating the validated document (Control Report Message Type is “Validated submission”). The MSA dispatch application disseminates the validated control report document IE717 (control report type is “Validated submission”) to the MSA destination application.

The MSA destination application performs structural validation on the received control report (IE717: C_CCR_DAT). If the control report indicates a request for a control, the MSA destination application flags the movement for performing a control at arrival of the goods.

The scenario of a control report submitted by the MSA of Dispatch is depicted in Figure 160 and Figure 161:

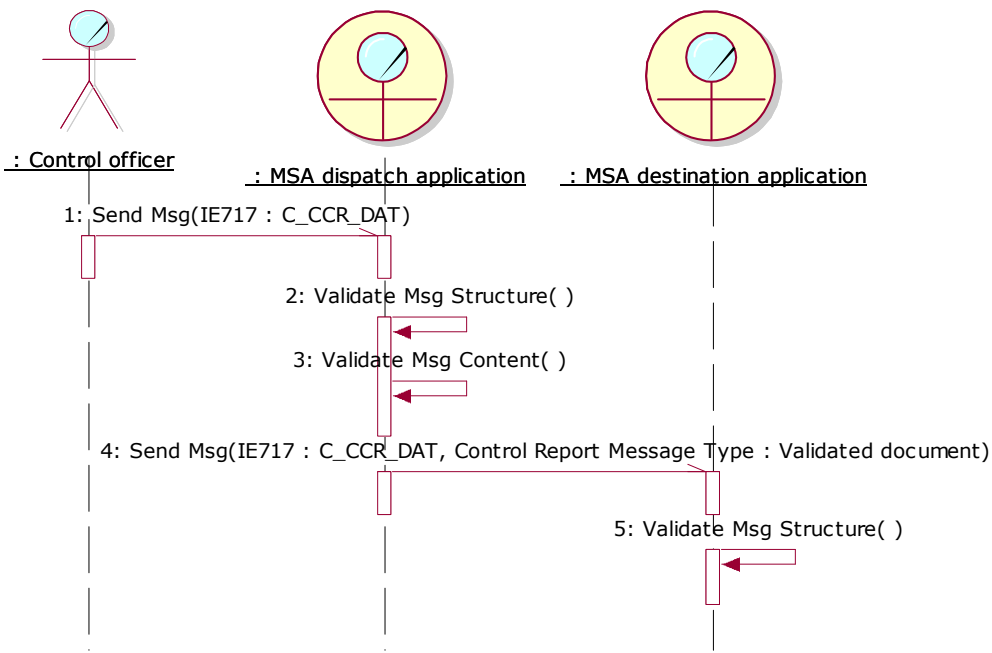


Figure 160: TSD - Control report submitted by the MSA of Dispatch

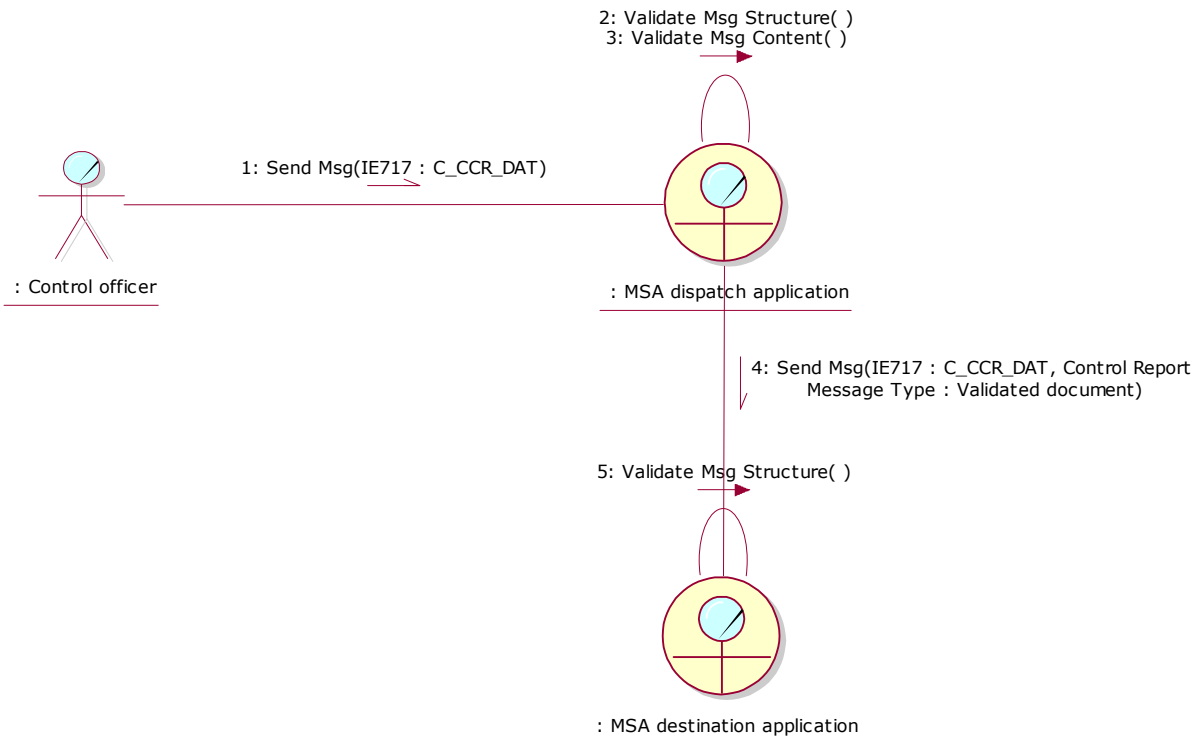


Figure 161: CLD - Control report submitted by the MSA of Dispatch

IV.I.2.2 Control report submitted by the MSA of Destination

This scenario assumes that the submitting Control officer resides at the MSA of Destination where the control occurred.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The Control officer submits his findings via a control report (IE717: C_CCR_DAT) to the MSA destination application, for validation and notification of the concerned MSAs. The draft control report is submitted without specifying a control report number.

The MSA destination application performs the relevant validations to the received control report (IE717: C_CCR_DAT), which pass successfully. If the MSA destination application has not received another control report prior to this reception, it assigns a unique reference number to the control report reference (the structure of control report reference and the Check Digit algorithm are defined in FESS Appendix B [A1]), otherwise it complements the existing report.

The MSA dispatch application is known to the MSA destination application, for disseminating the validated document (Control Report Message Type is “Validated submission”). The MSA of Destination should determine that it is synchronised with the MSA of Dispatch, thus confirming it is the current MSA of Destination rather than a former MSA of Destination.

The MSA destination application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that the MSA destination application has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA destination application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA destination application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA destination application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA destination application the IE934 (C_PAC_DAT) message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA destination application has all the information for the concerned movement via the IE934 message. It is indeed identified that the MSA destination application is the current MSA of Destination. Thus, the MSA destination application disseminates the validated control report document IE717 (control report type is “Validated submission”) to the MSA dispatch application.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The MSA dispatch application performs structural validation on the received control report (IE717: C_CCR_DAT). If the control report indicates a request for a control, the MSA dispatch application flags the movement for performing a control at arrival of the goods.

The scenario of a control report submitted by the MSA of Destination is depicted in Figure 162 and Figure 163:

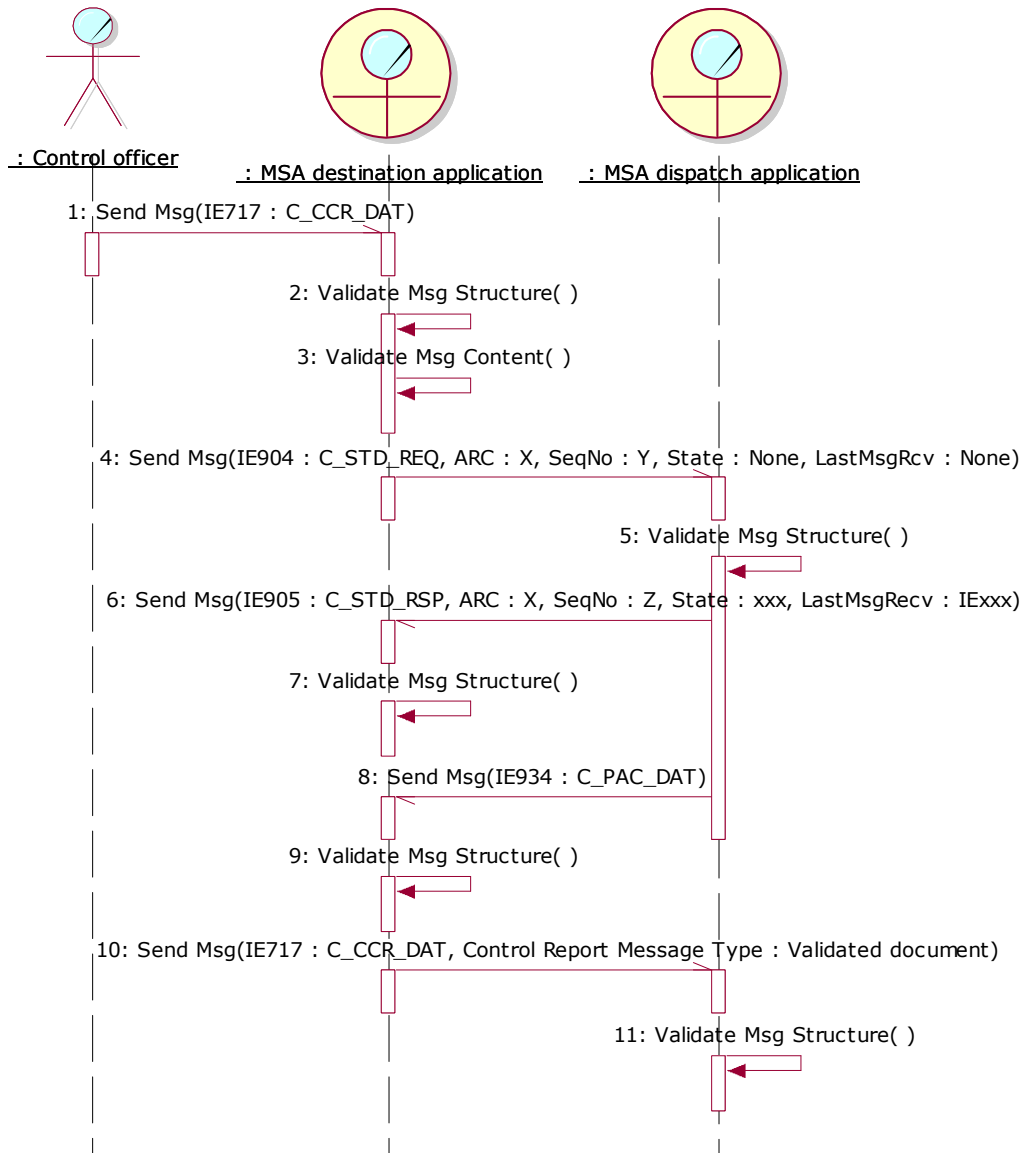


Figure 162: TSD – Control report submitted by the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

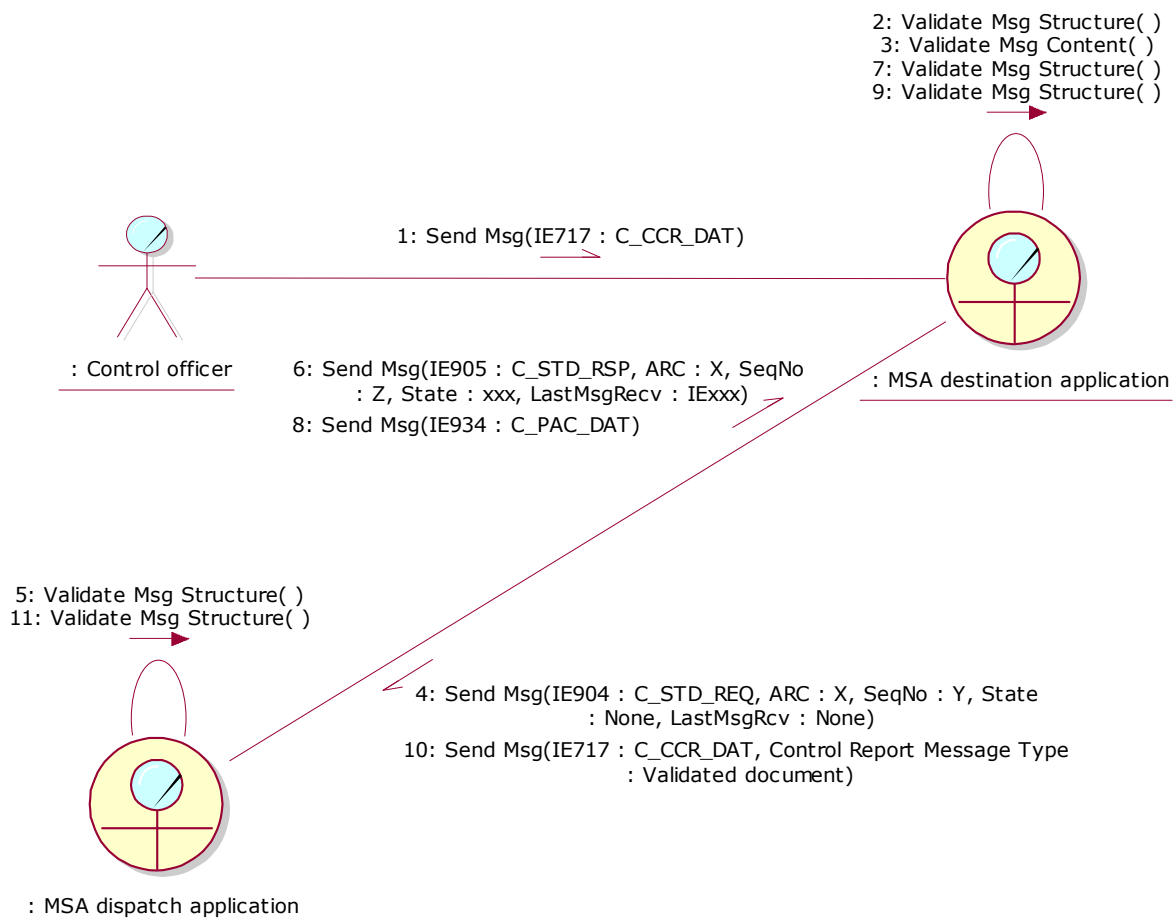


Figure 163: CLD - Control report submitted by the MSA of Destination

IV.I.2.3 Control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

This scenario assumes that the submitting Control officer resides at a third MSA where the control occurred, which is different from the MSA of Dispatch and the MSA of Destination.

The Control officer submits his findings via a control report (IE717: C_CCR_DAT) to his MSA application, a third MSA application, for validation and notification of the concerned MSAs. The draft control report is submitted without specifying a control report number.

The third MSA application performs the relevant validations to the received control report (IE717: C_CCR_DAT), which pass successfully. If the third MSA application has not received another control report prior to this reception, it assigns a unique reference number to the control report reference (the structure of control report reference and the Check Digit algorithm are defined in FESS Appendix B [A1]), otherwise it complements the existing report.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Since the third MSA application has not been involved in the movement (or has been involved in the movement as a former MSA of Destination), it has not received any IE messages concerning the movement (or has received some IE messages that do not reflect the current state of the movement). Thus, the MSA destination application is not known to the third MSA application for disseminating the validated document (Control Report Message Type is “Validated submission”).

The third MSA application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that in case that the third MSA application has not received any message that is related to the concerned ARC, then it has to set the “Status” Data Item to “None” and the “Last Received Message Type” Data Item to “None”, since the latter Data Item reflects the last message received from the MSA dispatch application. Furthermore, in case that the third MSA application is a former MSA of Destination with respect to the specific ARC, then it has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the third MSA application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the third MSA application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the third MSA application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the third MSA application the IE934 (C_PAC_DAT) message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the third MSA application has all the information for the concerned movement via the IE934 message. The third MSA application identifies the MSA of Destination and disseminates the validated control report document IE717 (control report type is “Validated submission”) to the MSA dispatch application and to the MSA destination application.

The MSA dispatch application validates the message structure of the received control report (IE717: C_CCR_DAT) and stores the message.

The MSA destination application performs structural validation on the received control report (IE717: C_CCR_DAT). If the control report indicates a request for a control, the MSA destination application flags the movement for performing a control at arrival of the goods.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The scenario for a control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 164 and Figure 165:

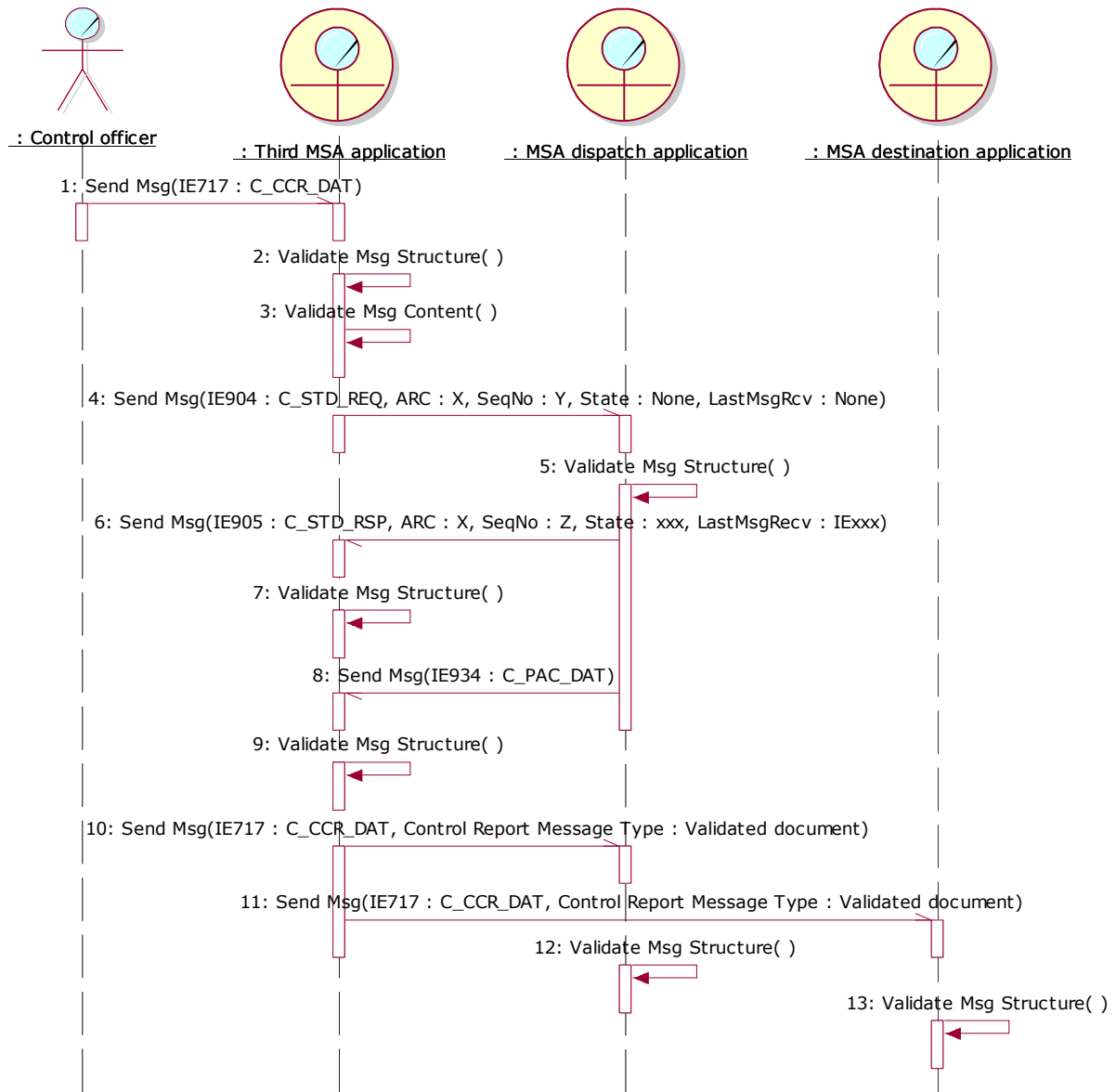


Figure 164: TSD - Control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

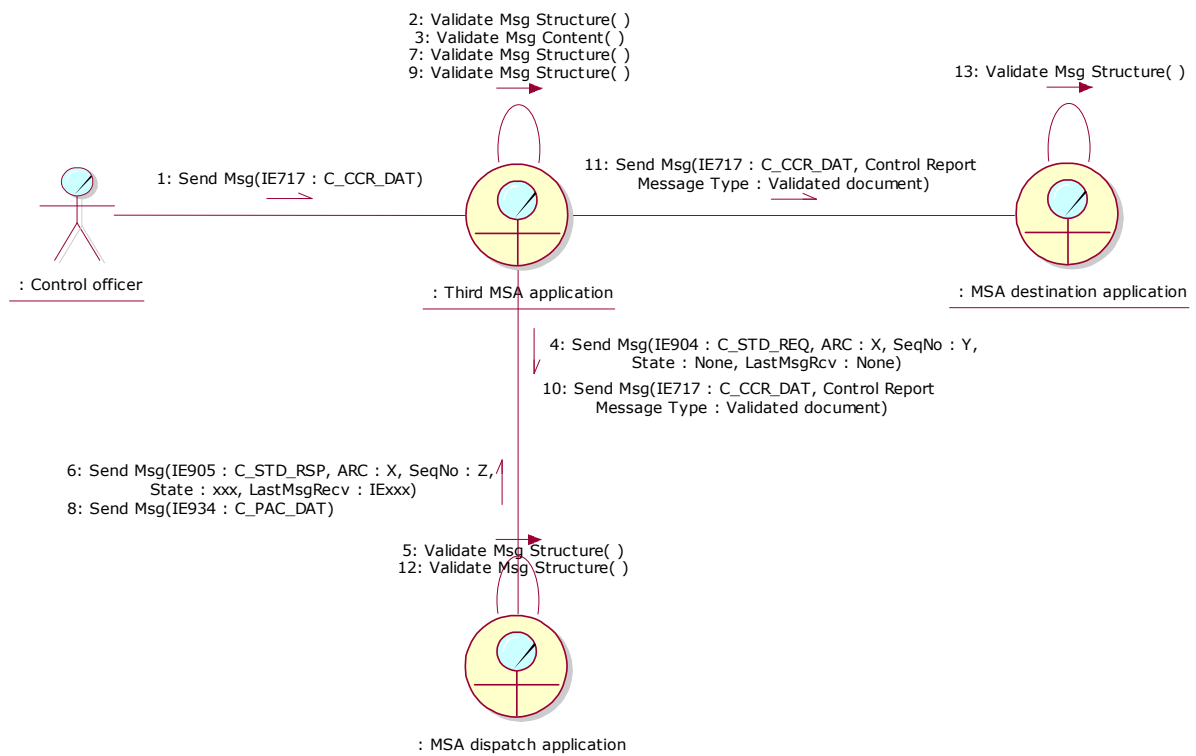


Figure 165: CLD - Control report submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

IV.I.3 Interruption of a movement (UC3.05)

There may be cases where an excise movement has been or needs to be immobilised before its normal completion. Interruptions are initiated by an Excise officer and it is assumed that a mechanism exists allowing the triggering of this use case. The details of this mechanism are a national matter.

A MSA is entitled to interrupt a movement if it considers that the movement cannot continue its journey to the destination or a serious offence has taken place in its territory.

The MSA of interruption in FESS [A1] is defined as the MSA where the interruption occurred, which also includes the MSA of Dispatch and the MSA of Destination. In the following scenarios, however, the MSA of interruption role is simplified and decomposed, hence wherever the MSA notifying the interruption is a third MSA (i.e. neither the MSA of Dispatch nor the MSA of Destination) the role third MSA will be utilised. The cases where the MSA of interruption is either the MSA of Dispatch or the MSA of Destination will be covered with the concrete roles (MSA of Dispatch or the MSA of Destination respectively) rather than the more abstract role MSA of interruption.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

Whenever the MSA notifying the interruption does not know all the addressees, it should perform a “Download of an e-AD” scenario (section III.I.12 Download of an e-AD (UC2.51)) in order to identify the addressees and disseminate the IE807. For example, when the interruption took place in a third MSA (that is, it is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.3.3) the interruption of movement message (IE807: C_STP_NOT) shall be disseminated to the MSA of Dispatch and the MSA of Destination. The MSA of Dispatch can be derived from the ARC of the movement (the third and fourth characters of the ARC denote the MSA of Dispatch). However, the current MSA of Destination will not be known. Therefore, the MSA of Destination will not become known unless a “Download of an e-AD” scenario is executed (see III.I.12 Download of an e-AD (UC2.51)).

The scenarios of this use case require the state of the concerned e-AD to be in “*Accepted*”, “*Rejected*”, “*Refused*”, or “*Partially refused*” state. In addition, the state of the concerned e-AD may be in the “*Exporting*” state.

The successful completion of the scenarios cause state transition at the MSA of Dispatch and the MSA of Destination for the concerned e-AD to “*Stopped*”. The state transition pertains to both the MSA of Dispatch and the MSA of Destination. Note that the state transition to “*Stopped*” state is included in the State-Transition Diagrams for Central Circuit Scenarios (see Sub-Section III.III) and in the State-Transition Diagrams for Export Scenarios (see Sub-Section III.IV).

The scenarios of this section describe the “Interruption of a movement (UC3.05)” use case, through the following scenarios:

- an Interruption of a movement submitted by the MSA of Dispatch (see IV.I.3.1);
- an Interruption of a movement submitted by the MSA of Destination (see IV.I.3.2);
- an Interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination (see IV.I.3.3).

IV.I.3.1 Interruption of a movement submitted by the MSA of Dispatch

This scenario assumes that an Excise Officer orders an interruption of a movement in the MSA dispatch application. In this case the MSA dispatch application, where the interruption occurred, notifies the MSA destination application by sending an interruption message (IE807: C_STP_NOT).

The MSA dispatch application transfers the movement interruption (IE807: C_STP_NOT) to the Consignor. If any timers associated with the interrupted e-AD have already expired at the limit date, the MSA dispatch application resets their flag that has been raised locally at expiration time. If any timers associated with the interrupted e-AD are still running, the MSA dispatch application stops them. Finally, the state of the movement at the MSA of Dispatch is set to “*Stopped*”.

The MSA destination application after validating the received message structure transfers, the movement interruption (IE807: C_STP_NOT) to the Consignee and sets the state of the movement at the MSA of Destination is set to “*Stopped*”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The scenario for an interruption of a movement submitted by the MSA of Dispatch is depicted in Figure 166 and Figure 167:

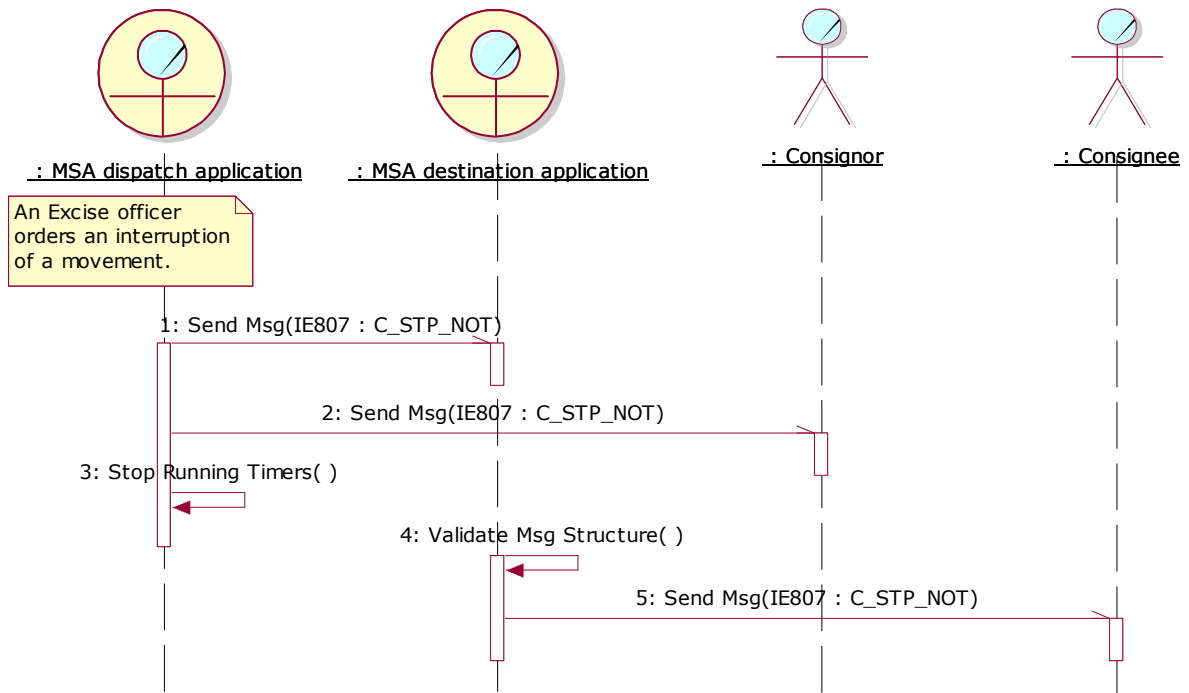


Figure 166: TSD – Interruption of a movement submitted by the MSA of Dispatch

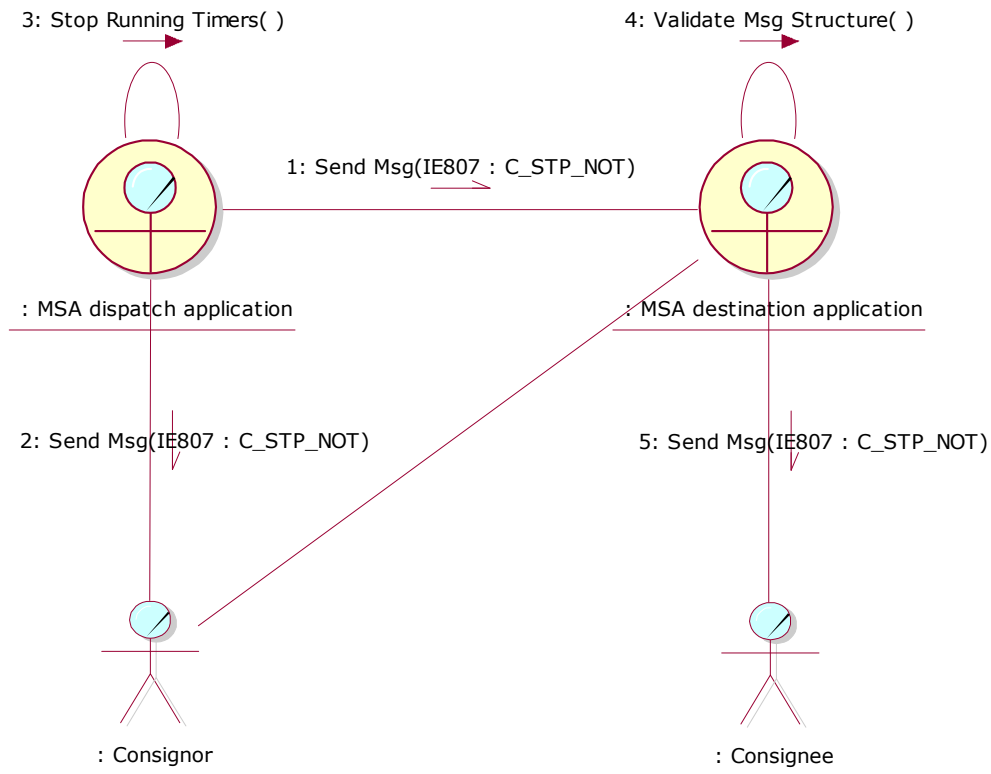


Figure 167: CLD - Interruption of a movement submitted by the MSA of Dispatch

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.I.3.2 Interruption of a movement submitted by the MSA of Destination

This scenario assumes that an Excise Officer orders an interruption of a movement in the MSA destination application.

The MSA dispatch application is known to the MSA destination application. The MSA of Destination should determine that the e-AD is in a state that can be interrupted and that it is synchronised with the MSA of Dispatch, thus confirming it is the current MSA of Destination rather than a former MSA of Destination.

The MSA destination application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request message (IE904: C_STD_REQ. For the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that the MSA destination application has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the MSA destination application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the MSA destination application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the MSA destination application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the MSA destination application the IE934 (C_PAC_DAT) message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the MSA destination application has all the information for the concerned movement via the IE934 message. It is indeed identified that the MSA destination application is the current MSA of Destination. In this case the MSA destination application, where the interruption occurred, notifies the MSA dispatch application by sending an interruption message (IE807: C_STP_NOT). The MSA destination application transfers the movement interruption (IE807: C_STP_NOT) to the Consignee. The state of the movement at the MSA of Destination is set to “*Stopped*”.

The MSA dispatch application after validating the received message structure transfers the movement interruption (IE807: C_STP_NOT) to the Consignor. If any timers associated with the interrupted e-AD have already expired at the limit date, the MSA dispatch application resets their flag that has been raised locally at expiration time. If any timers associated with the interrupted e-AD are still running, the MSA dispatch application stops them. Finally, the state of the movement at the MSA of Dispatch is set to “*Stopped*”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The scenario for an interruption of a movement submitted by the MSA of Destination is depicted in Figure 168 and Figure 169:

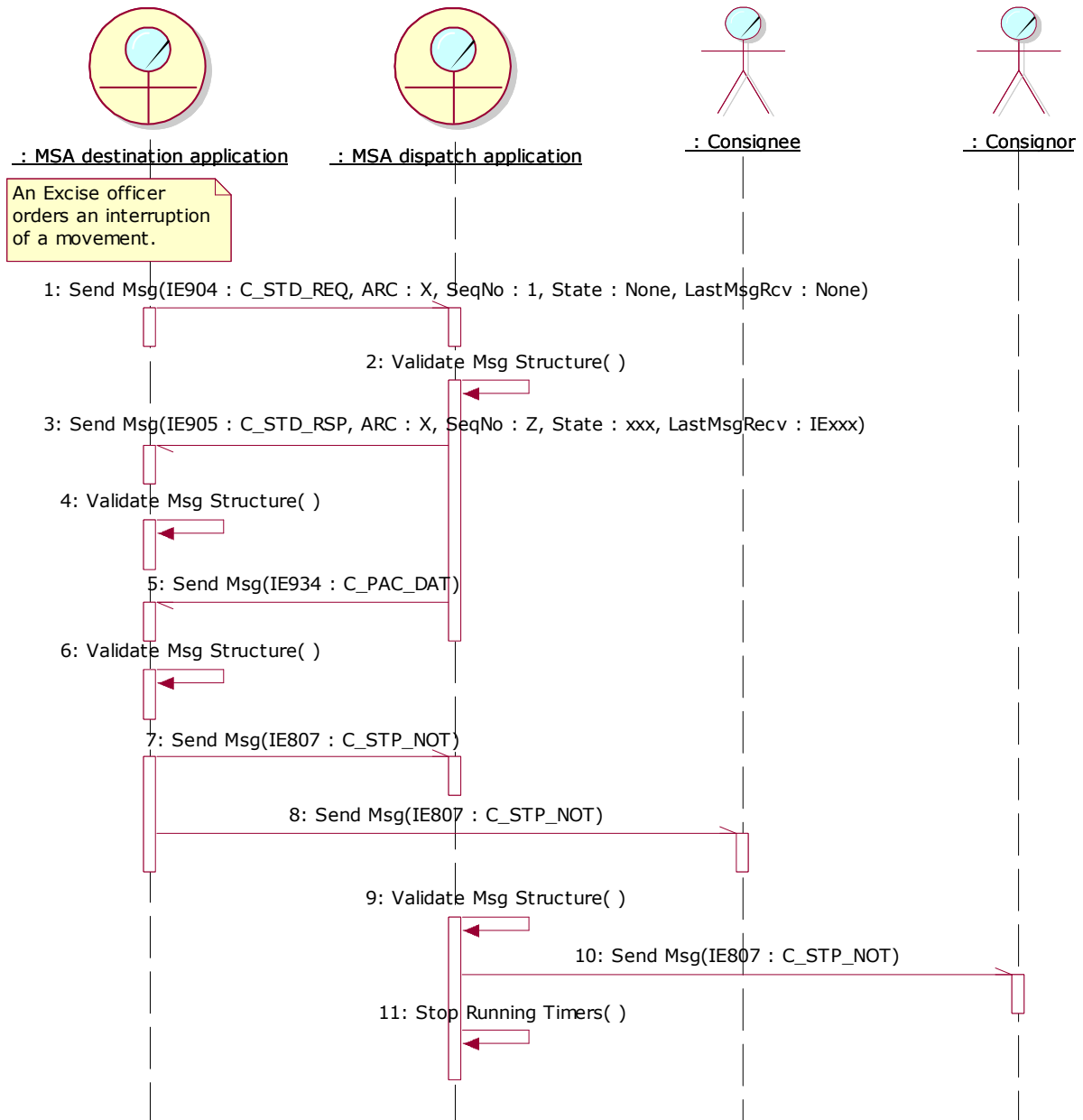


Figure 168: TSD – Interruption of a movement submitted by the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

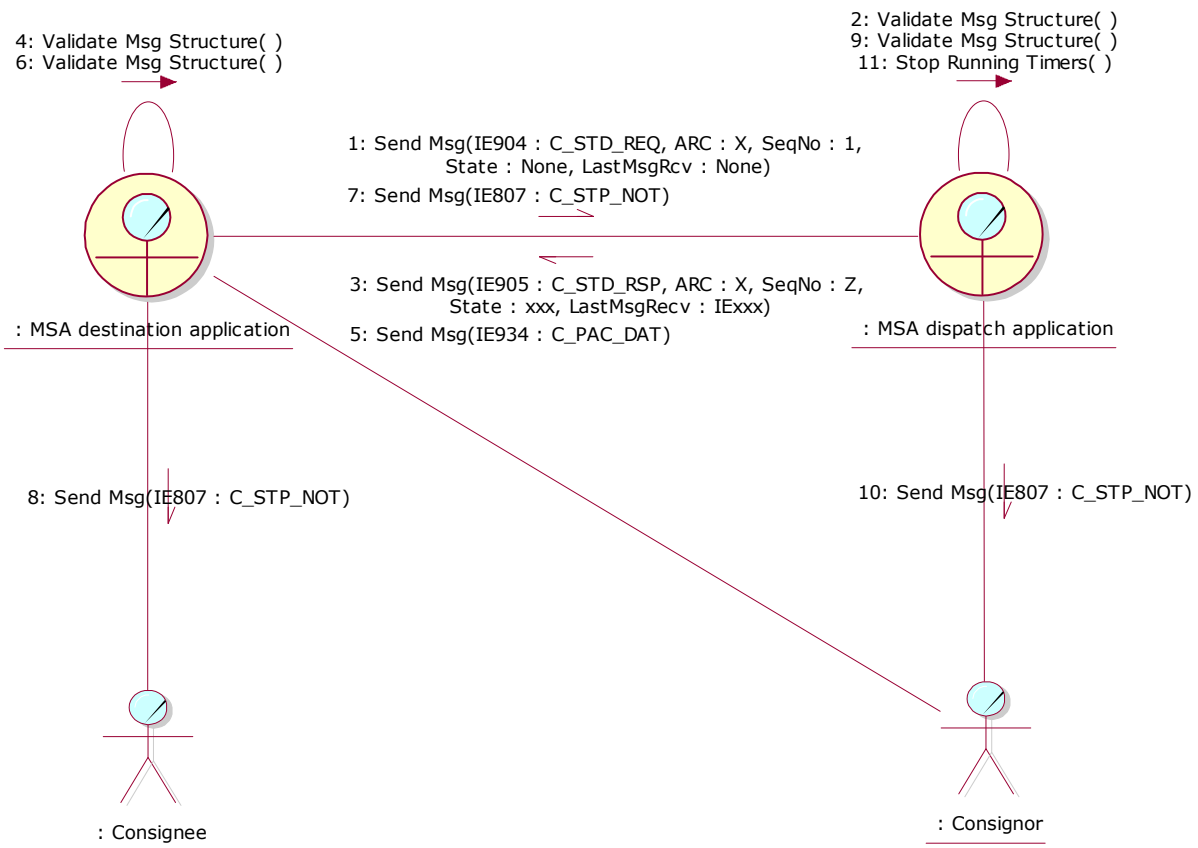


Figure 169: CLD - Interruption of a movement submitted by the MSA of Destination

IV.I.3.3 Interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

This scenario assumes that an Excise Officer orders an interruption of a movement in a third MSA application. The third MSA application, where the interruption occurred, notifies both the MSA dispatch application and the MSA destination application by disseminating an interruption message (IE807: C_STP_NOT).

Since the third MSA application has not been involved in the movement (or has been involved in the movement as a former MSA of Destination), it has not received any IE messages concerning the movement (or has received some IE messages that do not reflect the current state of the movement). Thus, the MSA destination application is not known to the third MSA application for disseminating the interruption message (IE807: C_STP_NOT). Moreover, in order to validate the correct state of the movement (being in “Accepted”, “Rejected”, “Refused”, “Partially refused”, or “Exporting” state) the third MSA application should consult the latest information for the e-AD by downloading its contents from the MSA of Dispatch (see section III.I.12 Download of an e-AD (UC2.51)). The IE905 message containing the “Status” Data Item of the movement will be utilised from the third MSA application, in order to check if the movement is in the correct status to be interrupted.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The third MSA application builds and sends to the MSA dispatch application (it is the MSA of which the Member State code appears in the third and fourth characters of the ARC) the Status Request (IE904: C_STD_REQ) message (for the correct values of the message see section III.I.12 Download of an e-AD (UC2.51)), including the ARC and the last known sequence number for the specific ARC, the status of the movement and the last message received from the MSA dispatch application. It is noted that in case that the third MSA application has not received any message that is related to the concerned ARC, then it has to set the “Status” Data Item to “None” and the “Last Received Message Type” Data Item to “None”, since the latter Data Item reflects the last message received from the MSA dispatch application. Furthermore, in case that the third MSA application is a former MSA of Destination with respect to the specific ARC, then it has to set the “Status” Data Item to the last known status and the “Last Received Message Type” Data Item to the last message received from the MSA dispatch application. Moreover, the third MSA application has to explicitly declare in sending the IE904: C_STD_REQ that it requests the full movement history.

The MSA dispatch application receives and validates the Status Request (IE904: C_STD_REQ) successfully (exceptional cases are described in Sub-Section III.VI). The MSA dispatch application identifies that the “Message Type” in the Status Request (IE904: C_STD_REQ) message is “2: Movement History Request”, indicating that the MSA Official at the third MSA application wishes to receive the full movement history.

In the case that the e-AD is found, the MSA dispatch application replies with a Status Response (IE905: C_STD_RSP) conveying the actual status of the movement (e.g. “Delivered”), the last known sequence number for the specific ARC and the last message received from the third MSA application for the specific ARC (e.g. “None”). Following the submission of the IE905, the MSA dispatch application also submits to the third application the IE934: C_PAC_DAT message that includes all business messages comprising the movement history.

Upon validating the structure of the received messages (IE905 and IE934) the third MSA application has all the information for the concerned movement via the IE934 message. The third MSA application identifies the MSA of Destination and disseminates the validated interruption message (IE807: C_STP_NOT) to the MSA dispatch application and to the MSA destination application.

The MSA dispatch application, after validating the received message structure, transfers the movement interruption (IE807: C_STP_NOT) to the Consignor. If any timers associated with the interrupted e-AD have already expired at the limit date, the MSA dispatch application resets their flag, which has been raised locally at expiration time. If any timers associated with the interrupted e-AD are still running, the MSA dispatch application stops them. Finally, the state of the movement at the MSA dispatch application is set to “*Stopped*”.

The MSA destination application, after validating the received message structure, transfers the movement interruption (IE807: C_STP_NOT) to the Consignee. Finally, the state of the movement at the MSA destination application is set to “*Stopped*”.

The scenario for an interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination is depicted in Figure 170 and Figure 171:

Section IV - Follow-up and collaboration

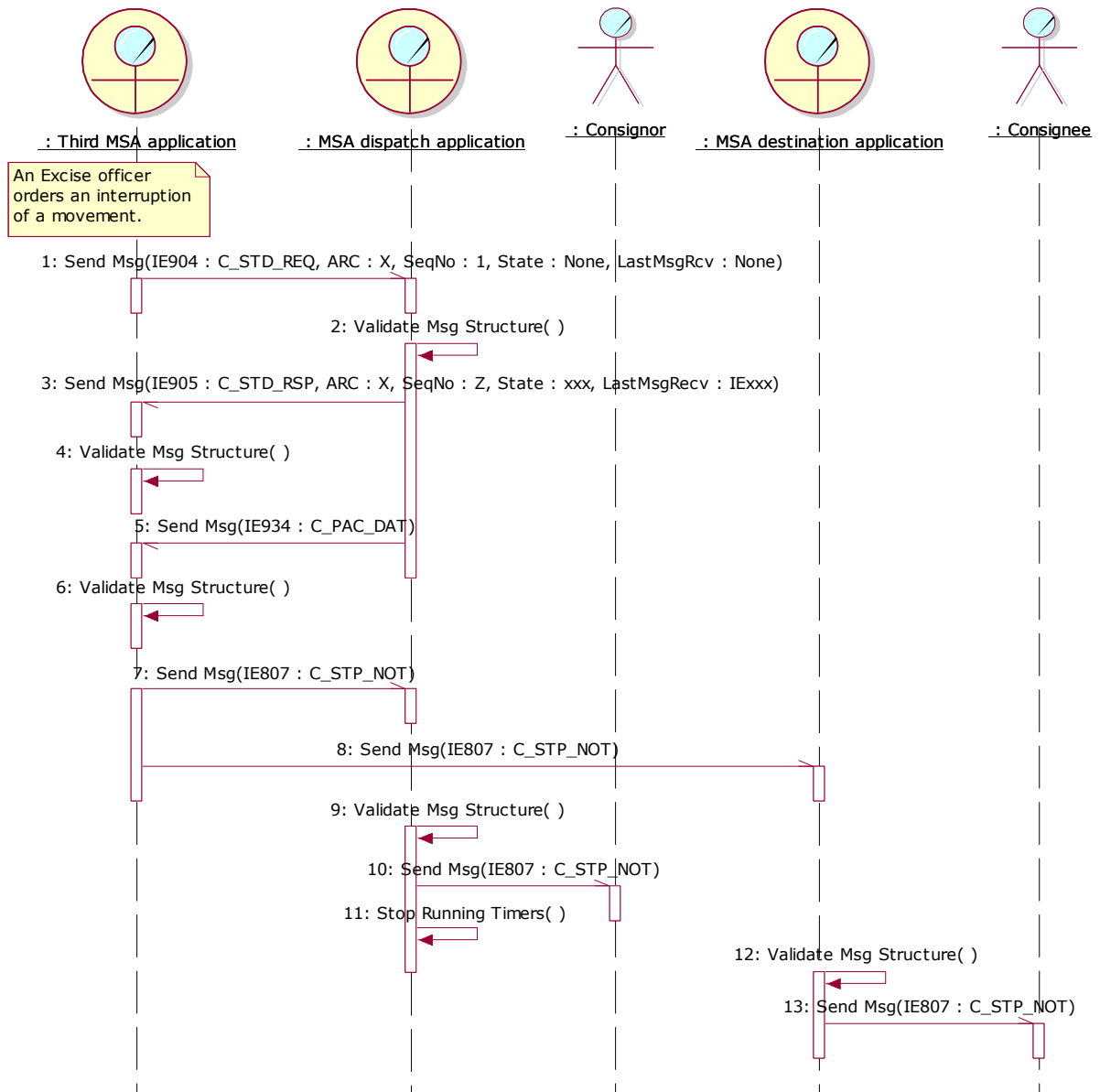


Figure 170: TSD – Interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

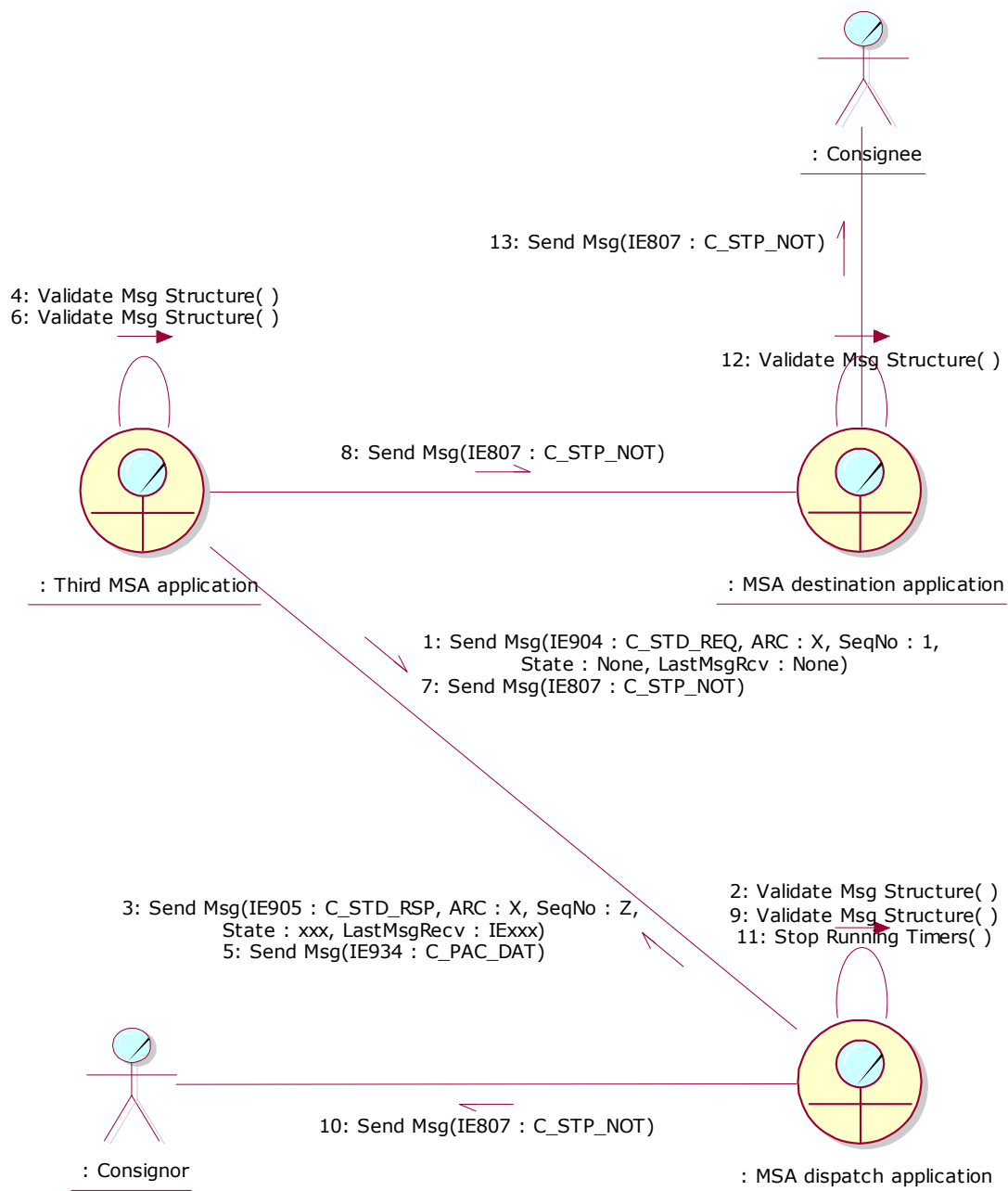


Figure 171: CLD - Interruption of a movement submitted by a third MSA that is neither the MSA of Dispatch nor the MSA of Destination

Sub-Section IV.II Administrative cooperation scenarios

IV.II.1 Spontaneous information (UC3.01)

The spontaneous information use case describes the exchange protocol between two MSA, the issuing and the addressed, for one-way information purposes only. No further actions are expected by the addressed MSA in terms of information exchange messages (excluding rejection messages in case of functional errors). The scenario does not describe how the Issuing MSA application is triggered in sending the message and it is assumed for a mechanism to exist (possibly a user interface).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The Issuing MSA application sends to the Addressed MSA application an administrative cooperation results (IE867: C_COO_RES) message. The Addressed MSA application validates the received message structure. The scenario for submitting an administrative cooperation results message is depicted in Figure 172 and Figure 173:

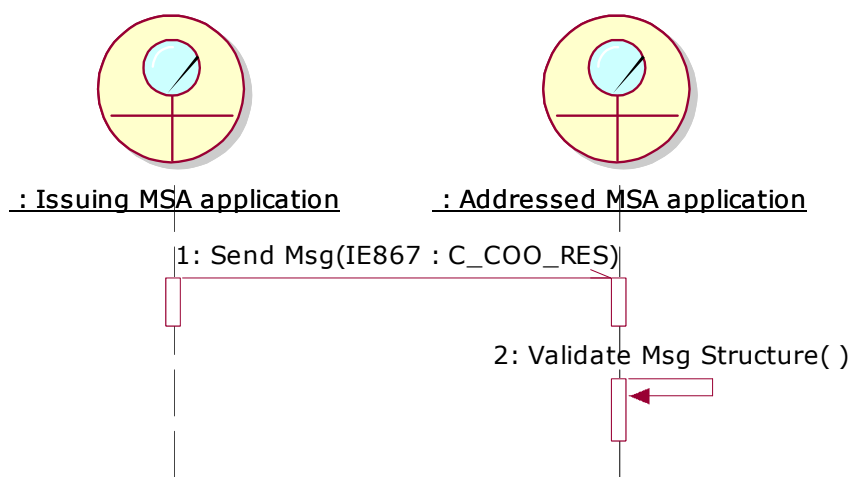


Figure 172: TSD - Spontaneous information (UC3.01)

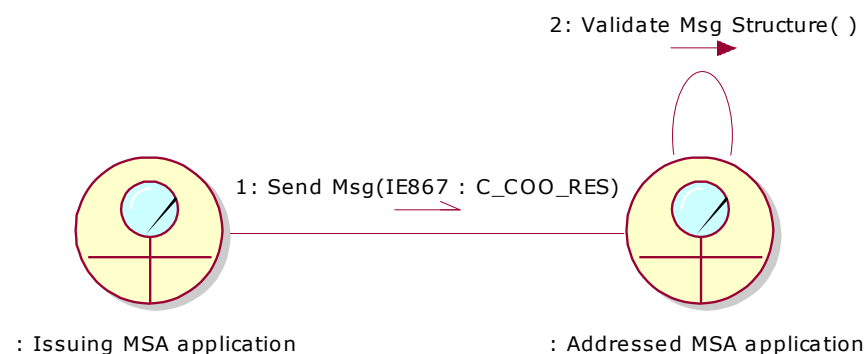


Figure 173: CLD - Spontaneous information (UC3.01)

IV.II.2 Request for assistance

The request for assistance use cases describe the exchange protocol between two MSAs, the requesting and the requested. Contrary to the spontaneous information use case, the receiver (i.e. the Requested MSA) is committed to respond with a results message. The request issued by the Requesting MSA contains a deadline for results that the Requested MSA may be extended by issuing an answer message.

The scenarios of this section describe the “Administrative cooperation - request for assistance (UC3.07)” and the “Administrative cooperation - deadline for results (UC3.09)” use cases, through the following scenarios:

- Submission of a request for assistance (see IV.II.2.1);
- Reminder at expiry time for sending the results to a request for assistance (see IV.II.2.2);

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

- Submission of an answer for extending the deadline to a request for assistance (see IV.II.2.3);
- Submission of the results to a request for assistance (see IV.II.2.4).

The request for assistance scenarios message exchanges cause state transitions, which are depicted in section IV.II.3 State-Transition Diagrams for Request for assistance scenarios.

IV.II.2.1 Submission of a request for assistance

This scenario describes the initiation of a request for assistance by the Requesting MSA. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included.

The Requesting MSA application submits an administrative cooperation request (IE721: C_COO_SUB) to the Requested MSA application indicating the Request Type as Administrative cooperation. The Requesting MSA application initiates the timer TIM_ACO to expire at the expected deadline for results as specified in the request message. The state of the request at the Requesting MSA is set to “Open”.

The Requested MSA application validates the received message structure and stores the message whilst setting the state to “Open” at the Requested MSA.

The scenario for the submission of a request for assistance is depicted in Figure 174 and Figure 175:

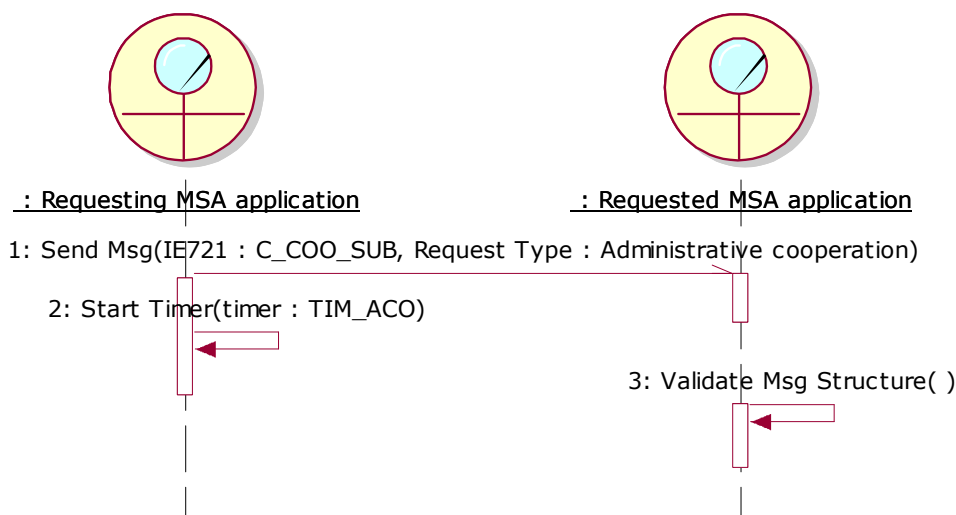


Figure 174: TSD - Submission of a request for assistance

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

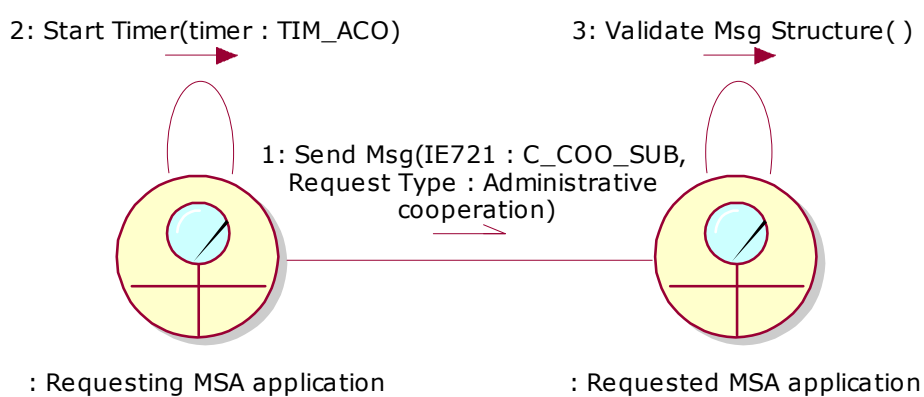


Figure 175: CLD - Submission of a request for assistance

IV.II.2.2 Reminder at expiry time for sending the results to a request for assistance

The scenario describes the case that the Requesting MSA has not received the results to a request for assistance within time and the timer TIM_ACO expires. The scenario requires that the concerned request is at the “Open” or “Extended” state.

Section	Scenario	Ending State
IV.II.2.1	Submission of a request for assistance	Open
IV.II.2.3	Submission of an answer for extending the deadline	Extended

Table 18: Scenarios that must have been previously completed in order to send a reminder for administrative cooperation

At the Requesting MSA application, the timer TIM_ACO expires without having received the results message (IE867: C_COO_RES) or an answer message (IE868: C_COO_ANS) for extending the deadline from the Requested MSA application. Alternatively, an answer message (IE868: C_COO_ANS) had been received from the Requested MSA application but the timer TIM_ACO, that was reset following the reception of the answer message, expires without having received the results message (IE867: C_COO_RES). The Requesting MSA application then sends a reminder message for administrative cooperation (IE869: C_COO_REM) to the Requested MSA application. The Requesting MSA application changes the state of the request from “Open” or “Extended” to “Late”.

The Requested MSA application validates the received message structure and stores the message whilst setting the state of the request from “Open” or “Extended” to “Late”.

The scenario for the expiry of the TIM_ACO timer is depicted in Figure 176 and Figure 177:

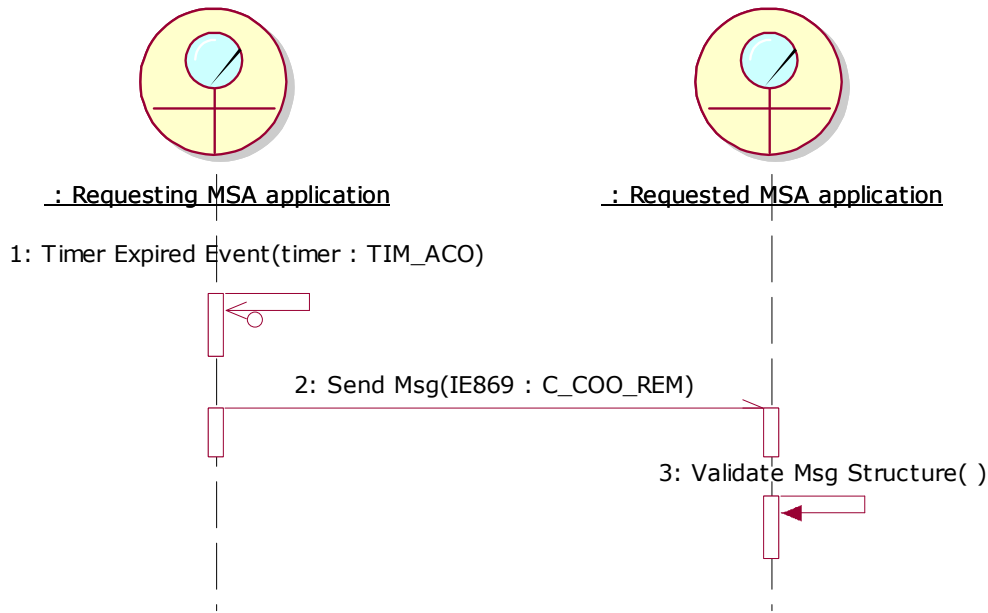


Figure 176: TSD - Reminder at expiry time for sending the results to a request for assistance

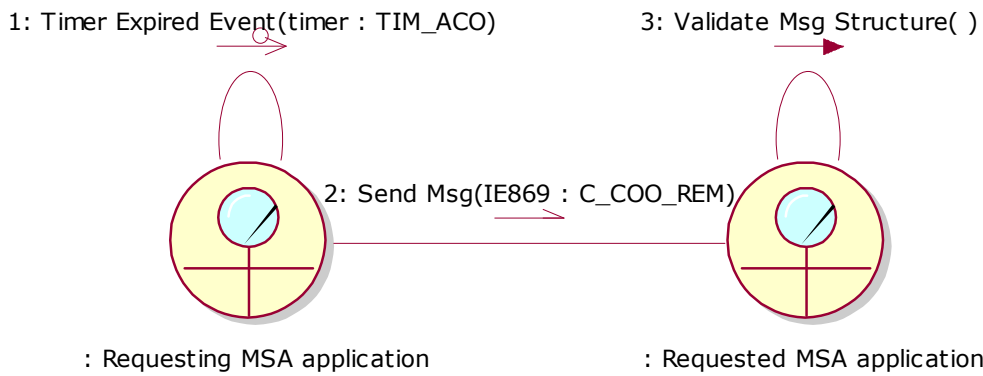


Figure 177: CLD - Reminder at expiry time for sending the results to a request for assistance

IV.II.2.3 Submission of an answer for extending the deadline to a request for assistance

The scenario describes the case that the Requested MSA sends an answer for extending the deadline for providing the results to a request for assistance. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires that the concerned request is at the “Open” or “Late” state.

Section	Scenario	Ending State
IV.II.2.1	Submission of a request for assistance	Open
IV.II.2.2	Reminder at expiry time for sending the results to a request for assistance	Late

Table 19: Scenarios that must have been previously completed in order to submit an answer to a request for assistance

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The ELO of the Requested MSA decides to extend the deadline to a request for assistance in order to complete the requested actions for a given reason. The Requested MSA application sends an answer message (IE868: C_COO_ANS) and updates the state of the request from “Open” or “Late” to “Extended”.

The Requesting MSA application validates the received message structure and updates the state of the request from “Open” or “Late” to “Extended”. Then it resets the timer TIM_ACO to expire at “Deadline for Results” date and time as specified by the answer message (IE868: C_COO_ANS). In addition if the timer TIM_ACO has expired the Requesting MSA application clears the flag that has been raised locally.

The scenario for submission of an answer is depicted in Figure 178 and Figure 179:

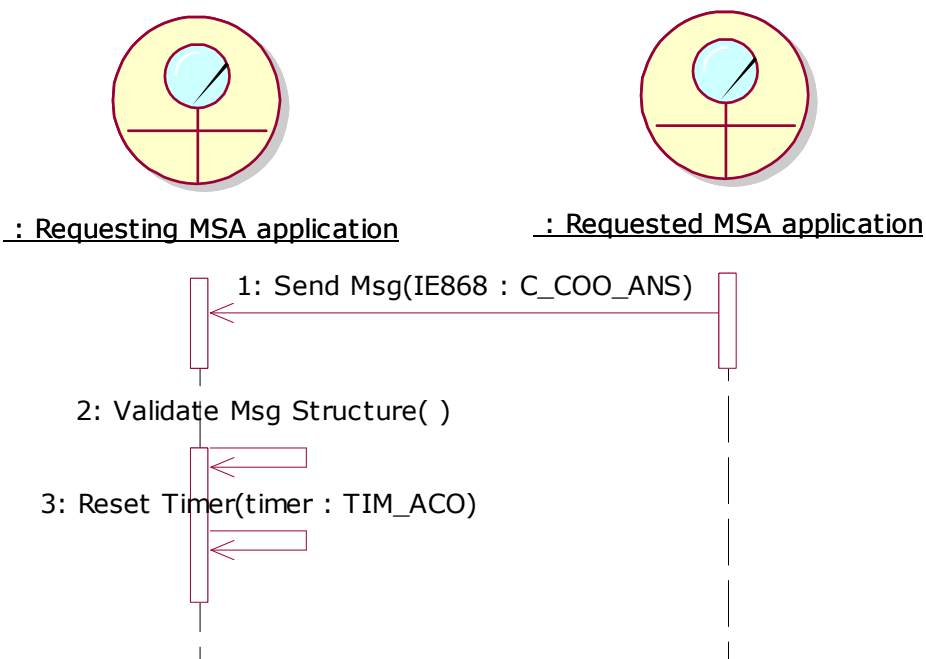


Figure 178: TSD - Submission of an answer for extending the deadline to a request for assistance

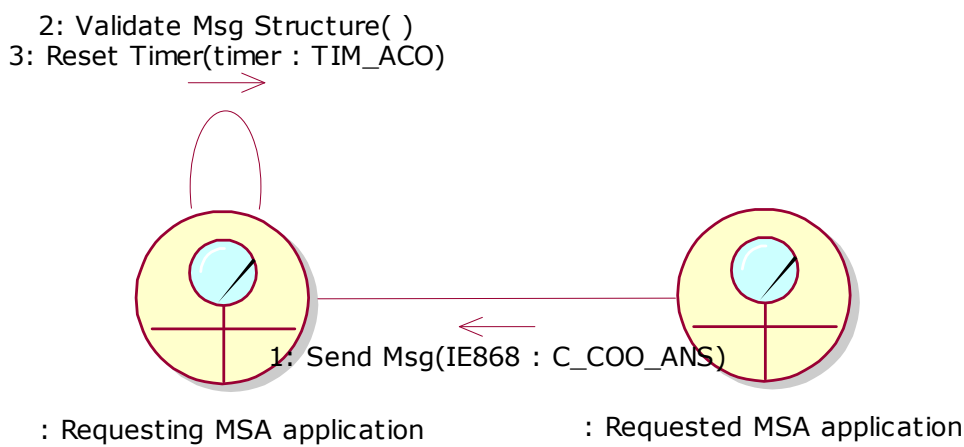


Figure 179: CLD - Submission of an answer for extending the deadline to a request for assistance

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.II.2.4 Submission of the results to a request for assistance

The scenario describes the case that the Requested MSA sends the results to a request for assistance. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires the concerned request to be in the “Open”, “Extended” or “Late” state.

Section	Scenario	Ending State
IV.II.2.1	Submission of a request for assistance	Open
IV.II.2.2	Reminder at expiry time for sending the results to a request for assistance	Late
IV.II.2.3	Submission of an answer for extending the deadline to a request for assistance	Extended

Table 20: Scenarios that must have been previously completed in order to submit the results to a request for assistance

The ELO of the Requested MSA completes the requested actions specified by the initial received administrative cooperation request (IE721: C_COO_SUB) message from the requesting MSA. The results of these actions (IE867: C_COO_RES) are then sent by the Requested MSA application to the Requesting MSA application. Finally, the Requested MSA application sets the state of the request from “Open”, “Extended” or “Late” to “Closed”.

The Requesting MSA validates the received administrative cooperation results (IE867: C_COO_RES) structure and changes the state of the request from “Open”, “Extended” or “Late” to “Closed”. Finally, the Requesting MSA stops the TIM_ACO timer.

The scenario for request for assistance with TIM_ACO expiration and reception of results is depicted in Figure 180 and Figure 181:

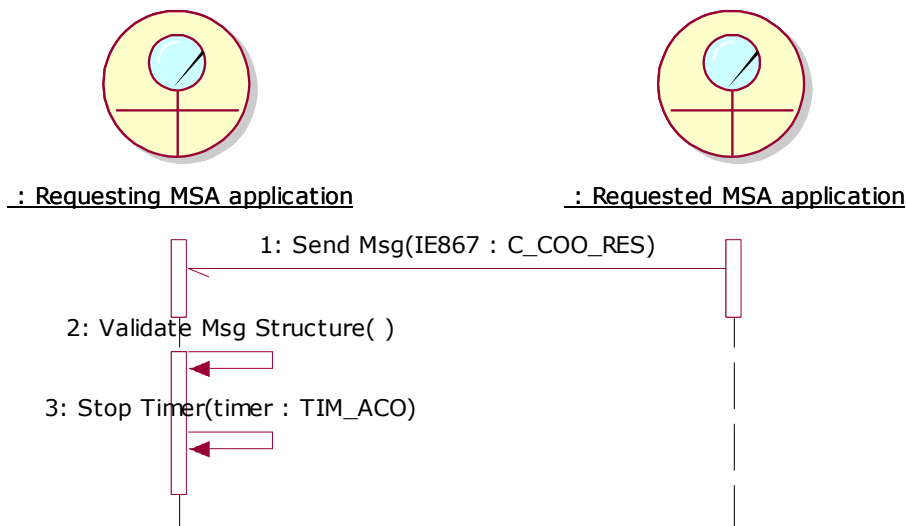


Figure 180: TSD - Submission of the results to a request for assistance

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

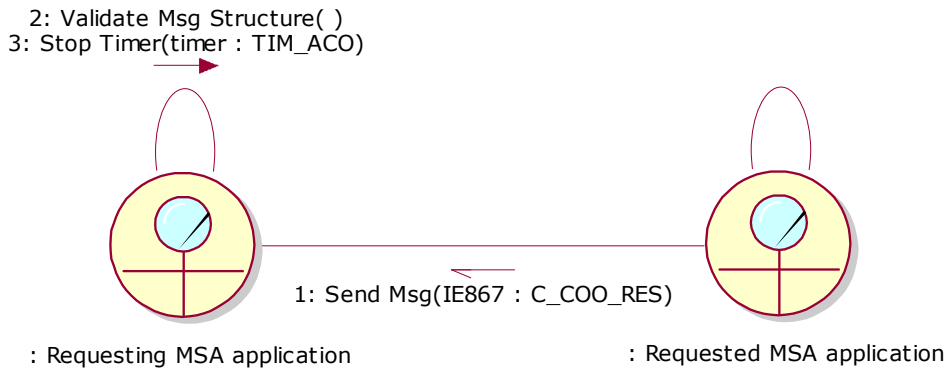


Figure 181: CLD - Submission of the results to a request for assistance

IV.II.3 State-Transition Diagrams for Request for assistance scenarios

IV.II.3.1 Requesting MSA

The following State Transition Diagram in Figure 196 depicts the State Machine at the Requesting MSA. All the transitions are described in detail in Sub-Section IV.II.

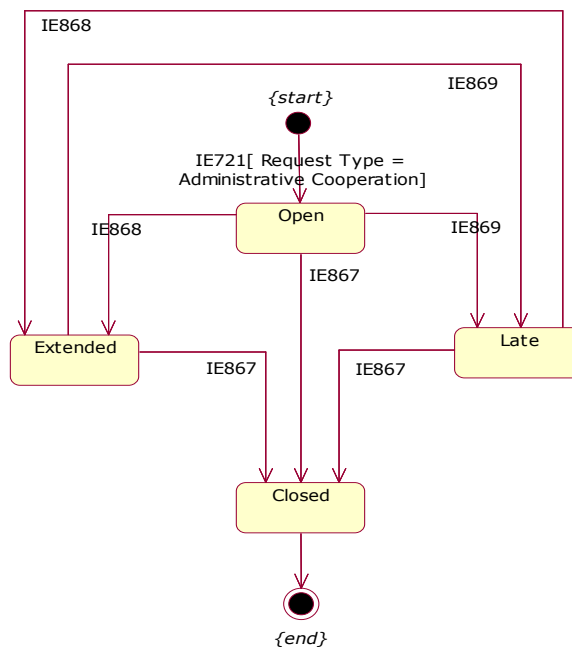


Figure 182: STD at Requesting MSA for Request for assistance scenarios

IV.II.3.2 Requested MSA

The following State Transition Diagram in Figure 197 depicts the State Machine at the Requested MSA. All the transitions are described in detail in Sub-Section IV.II.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

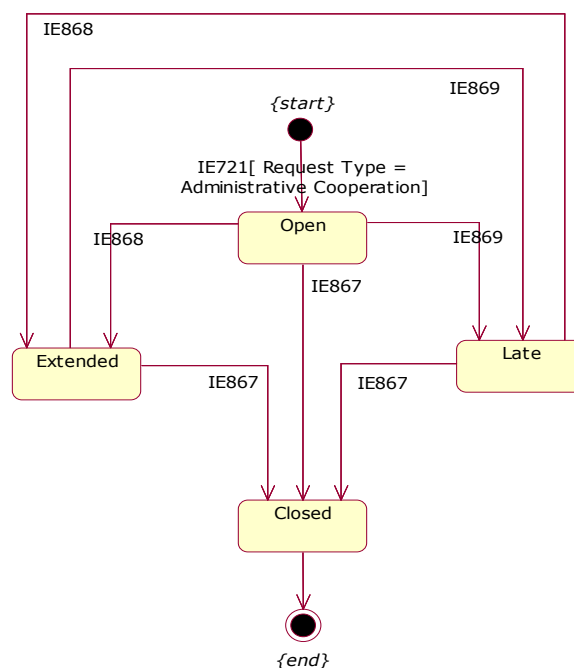


Figure 183: STD at Requested MSA for Request for assistance scenarios

Sub-Section IV.III Access to the history of movements

IV.III.1 Request for history information

The scenarios of this section describe the consultation of offline e-AD information due to the e-AD not being available on-line anymore (i.e. the e-AD exists but it has been archived). It should be noted that the requests for history information concern only ELO to ELO requests. The requesting ELO may have previously performed an e-AD download that failed (as described in section III.I.12.2) due to the e-AD not being available on-line anymore. However, a request for history information is not necessarily preceded by an e-AD download that failed. Similar to the request for assistance scenarios (see section IV.II.2), the request for history information scenarios cause state transitions, which are depicted in section IV.III.2 State-Transition Diagrams for Request for history scenarios.

The scenarios of this section describe the “Request for history information (UC3.29)” and the “History - deadline for results (UC3.14)” use cases, through the following scenarios:

- Submission of a history information request (see IV.III.1.1);
- Reminder at expiry time for sending the results to a history information request (see IV.III.1.2);
- Submission of an answer for extending the deadline to a history information request (see IV.III.1.3);
- Submission of an answer for refusal of a history information request (see IV.III.1.4);
- Submission of the results to a history information request (see IV.III.1.5);

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

- Submission of partial results to a history information request (see IV.III.1.6).

The request for history information message exchanges cause state transitions, which are depicted in section IV.III.2 State-Transition Diagrams for Request for history scenarios.

It should be noted that the data that can be included in a history results message (IE820: C_HIM_RES) is:

- Any applicable core business message concerning the requested ARC (as in the case of comprising the history of a movement via an IE934);
- Any IE721 message concerning the requested ARC but only with <Request Type> = "1: Administrative cooperation";
- Any IE867 message concerning the requested ARC which might either represent ACO Results or Spontaneous Information.

Concerning the coordination protocol validations, invalid values of “Follow Up Correlation ID” should be refused via an IE906 message (refer to VIII.I.3.2.3.1 Coordination protocol validations).

IV.III.1.1 Submission of a history information request

This scenario describes the initiation of a history information request (IE721: C_COO_SUB) by the Requesting MSA. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included.

The Requesting MSA application submits an administrative cooperation request (IE721: C_COO_SUB) to the Requested MSA application indicating the Request Type as Request for history. The Requesting MSA application initiates the timer TIM_HIS to expire at the expected deadline for results as specified in the request message. The state of the request at the Requesting MSA is set to “Open”.

The Requested MSA application validates the received message structure, stores the message, and sets the state to “Open”.

The scenario for request for history information is depicted in Figure 184 and Figure 185:

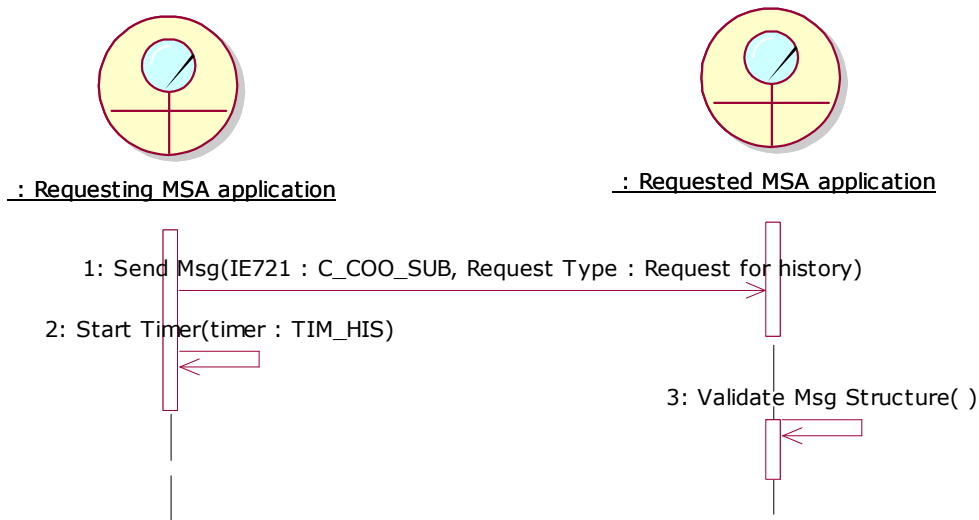


Figure 184: TSD - Submission of a history information request

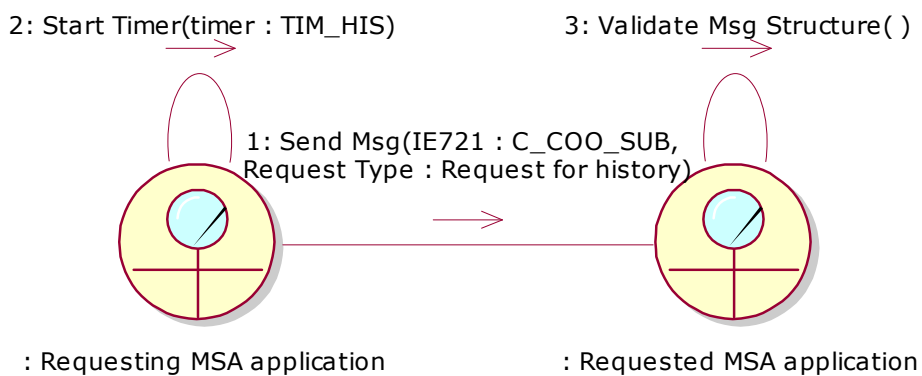


Figure 185: CLD- Submission of a history information request

IV.III.1.2 Reminder at expiry time for sending the results to a history information request

The scenario describes the case that the Requesting MSA has not received the results to a history information request within time and the timer TIM_HIS expires. The scenario requires that the concerned request is at the “Open” or “Extended” state.

Section	Scenario	Ending State
IV.III.1.1	Submission of a history information request	Open
IV.III.1.3	Submission of an answer for extending the deadline to a history information request	Extended

Table 21: Scenarios that must have been previously completed in order to send a reminder for history information

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The timer TIM_HIS expires without having received a history results message (IE820: C_HIM_RES) or an answer message (IE868: C_COO_ANS) for extending the deadline from the Requested MSA application. Alternatively, an answer message (IE868: C_COO_ANS) for extending the deadline had been received from the Requested MSA application but the timer TIM_HIS, that was reset following the reception of the answer message, expires without having received a history results message (IE820: C_HIM_RES). The Requesting MSA application then sends a reminder message for history information (IE869: C_COO_REM) to the Requested MSA application. The Requesting MSA application changes the state of the concerned request from “Open” or “Extended” to “Late”.

The Requested MSA application validates the received message structure and stores the message whilst setting the state from “Open” or “Extended” to “Late” at the Requested MSA.

The scenario for the expiry of the TIM_HIS timer is depicted in Figure 186 and Figure 187:

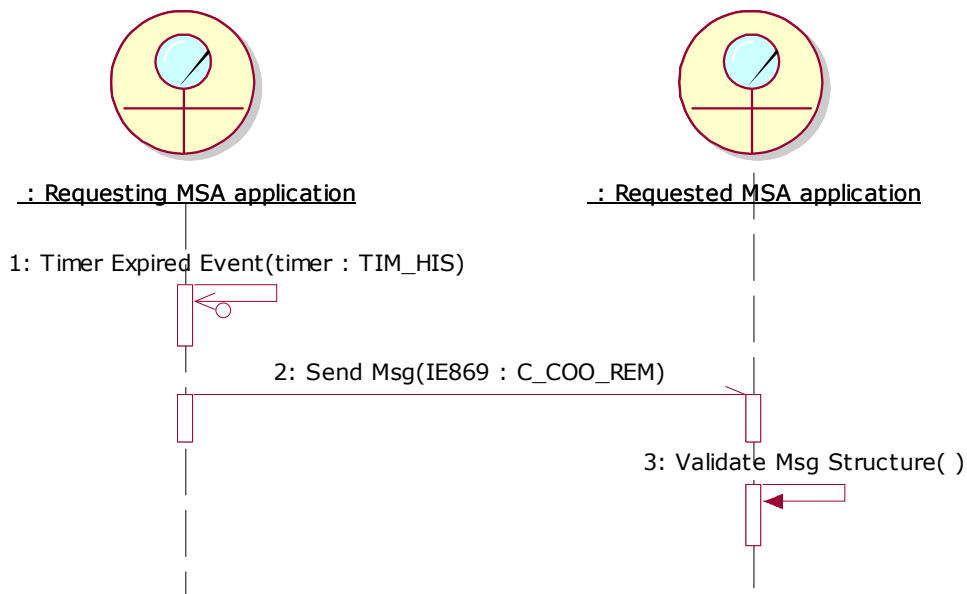


Figure 186: TSD - Reminder at expiry time for sending the results to a request for assistance

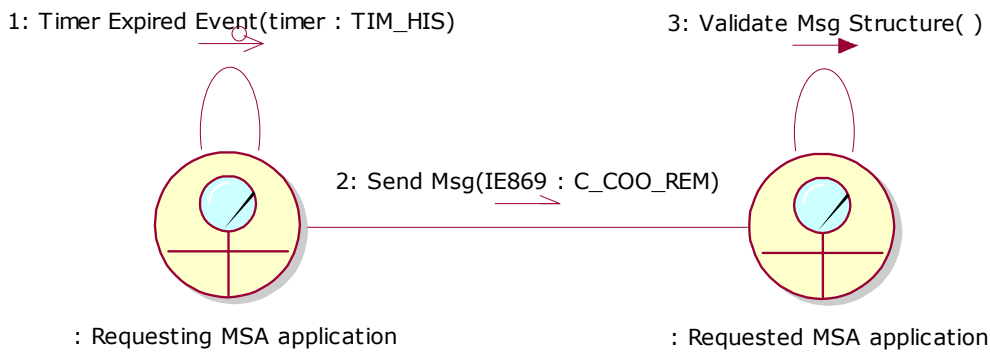


Figure 187: CLD - Reminder at expiry time for sending the results to a request for assistance

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.III.1.3 Submission of an answer for extending the deadline to a history information request

The scenario describes the case that the Requested MSA sends an answer for extending the deadline for providing the requested information to a history information request. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires that the concerned request is at the “Open” or “Late” state.

Section	Scenario	Ending State
IV.III.1.1	Submission of a history information request	Open
IV.III.1.2	Reminder at expiry time for sending the results to a history information request	Late

Table 22: Scenarios that must have been previously completed in order to submit an answer for extending the deadline to a history information request

The ELO of the Requested MSA decides to extend the deadline to a history information request in order to complete the requested history information for a given reason. The Requested MSA application sends an answer message (IE868: C_COO_ANS) with no History Refusal Reason Code. The state of the request at the Requested MSA is set from “Open” or “Late” to “Extended”.

The Requesting MSA application validates the received message structure and updates the state of the request from “Open” or “Late” to “Extended”. The Requesting MSA application then resets the timer TIM_HIS to expire at “Deadline for Results” date and time as specified by the answer message (IE868: C_COO_ANS). In addition if the timer TIM_HIS has expired it clears the flag that has been raised locally.

The scenario for submission of an answer for extending the deadline to a history information request is depicted in Figure 188 and Figure 189:

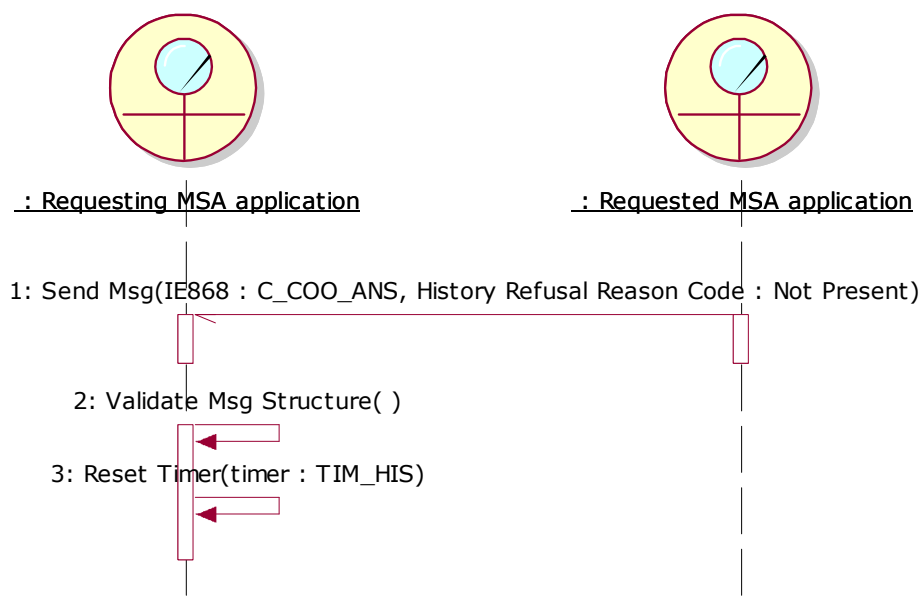


Figure 188: TSD - Submission of an answer for extending the deadline to a history information request

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

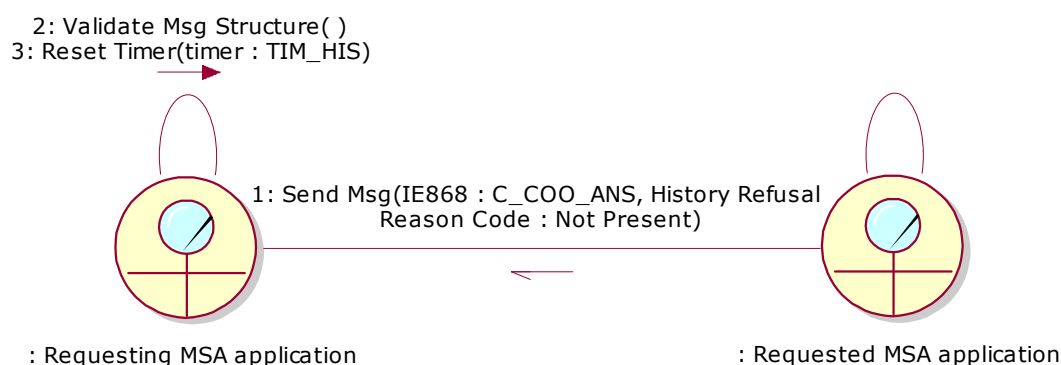


Figure 189: CLD - Submission of an answer for extending the deadline to a history information request

IV.III.1.4 Submission of an answer for refusal of a history information request

The scenario describes the case that the Requested MSA sends an answer for refusing a history information request. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires that the concerned request is at the “Open” state.

Section	Scenario	Ending State
IV.III.1.1	Submission of a history information request	Open

Table 23: Scenarios that must have been previously completed in order to submit an answer for refusing a history information request

The ELO of the Requested MSA examines the contents of the request and before the expiration of the deadline decides to refuse the request for history information for a given reason. The Requested MSA application sends an answer message (IE868: C_COO_ANS) with a specific History Refusal Reason Code. The state of the request at the Requested MSA is set from “Open” to “Closed”.

The Requesting MSA application validates the received message structure, while updating the state of the request at the Requesting MSA from “Open” to “Closed”. The Requesting MSA stops the TIM_HIS timer.

The scenario for submission of an answer for resusing a history information request is depicted in Figure 188 and Figure 189:

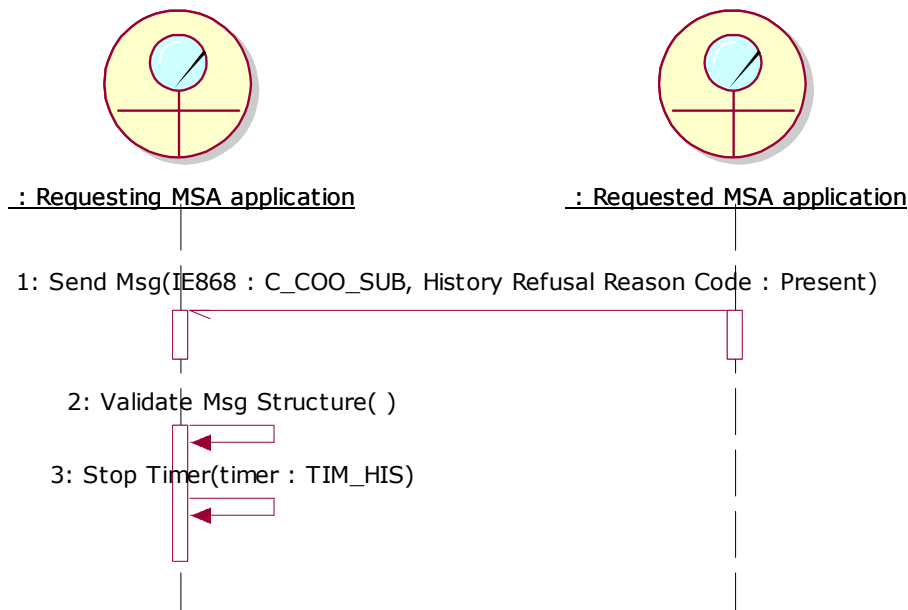


Figure 190: TSD - Submission of an answer for refusal of a history information request

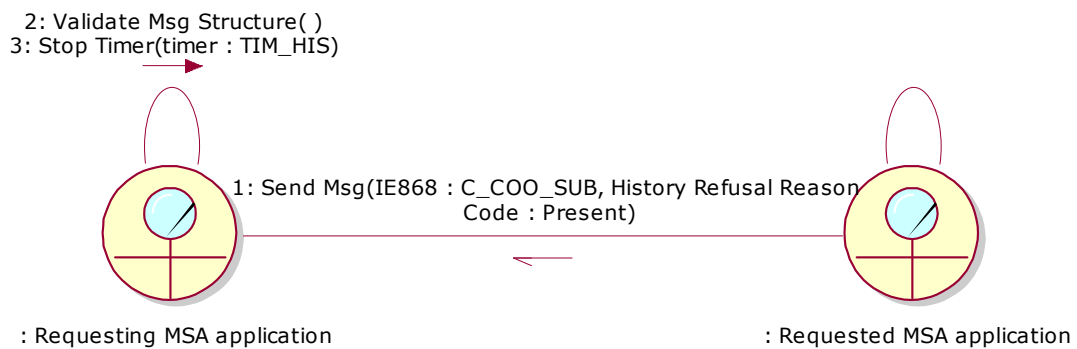


Figure 191: CLD - Submission of an answer for refusal of a history information request

IV.III.1.5 Submission of the results to a history information request

The scenario describes the case that the Requested MSA sends the results to a history information request. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires the concerned request to be in the “Open”, “Extended”, “Late” or “Partially answered” state.

Section	Scenario	Ending State
IV.III.1.1	Submission of a history information request	Open
IV.III.1.2	Reminder at expiry time for sending the results to a history information request	Late
IV.III.1.3	Submission of an answer for extending the deadline to a history information request	Extended
IV.III.1.6	Submission of partial results to a history information request	Partially answered

Table 24: Scenarios that must have been previously completed in order to submit the results to a history information request

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The ELO of the Requested MSA completes the requested actions specified by the initial received history information request from the requesting MSA. The results of these actions are then sent by the Requested MSA application to the Requesting MSA application via a history results message (IE820: C_HIM_RES). The history results message (IE820: C_HIM_RES) indicates that these are the final results by having the <Complement of Information Expected flag> Data Item equal to “0”. Finally, the Requested MSA application sets the state at the Requested MSA from “Open”, “Extended”, “Late” or “Partially answered” to “Closed”.

The Requesting MSA application validates the received history results (IE820: C_HIM_RES) structure and checks that the <Complement of Information Expected flag> Data Item is “0”. The Requesting MSA application changes the state at the Requesting MSA to “Closed”. Finally, the Requesting MSA application stops the TIM_HIS timer, if it is running.

The scenario for the submission of results to a history information request is depicted in Figure 192 and Figure 193:

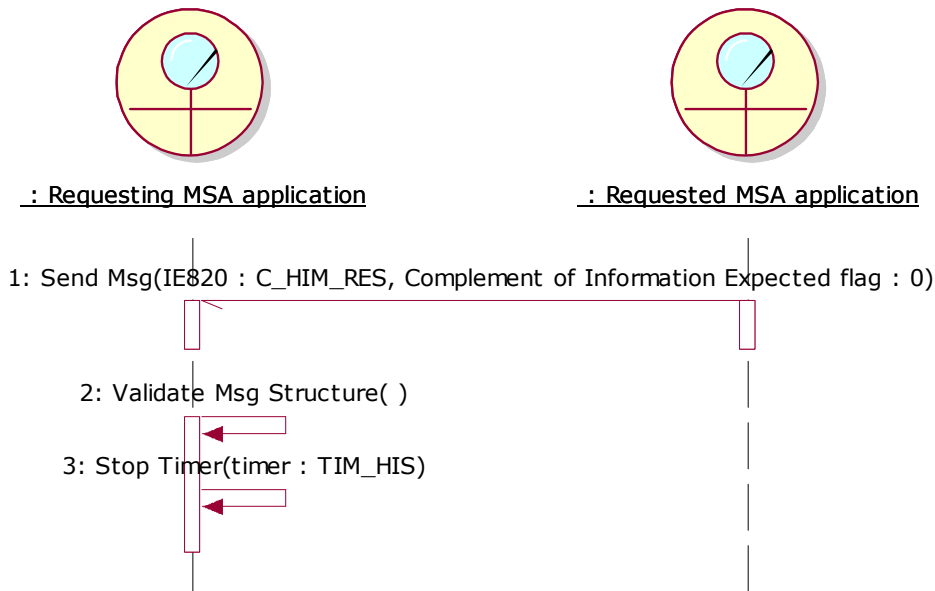


Figure 192: TSD - Submission of the results to a history information request

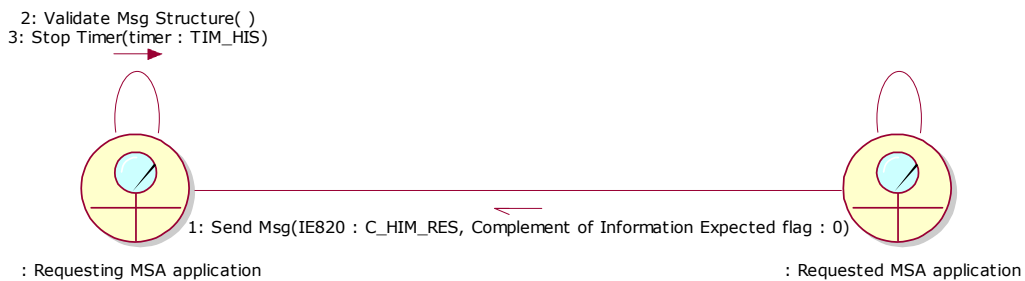


Figure 193: CLD - Submission of the results to a history information request

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.III.1.6 Submission of partial results to a history information request

The scenario describes the case that the Requested MSA sends partial results to a history information request. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires the concerned request to be in the “*Open*”, “*Extended*”, “*Late*” or “*Partially answered*” state.

Section	Scenario	Ending State
IV.III.1.1	Submission of a history information request	Open
IV.III.1.2	Reminder at expiry time for sending the results to a history information request	Late
IV.III.1.3	Submission of an answer for extending the deadline to a history information request	Extended
IV.III.1.6	Submission of partial results to a history information request	Partially answered

Table 25: Scenarios that must have been previously completed in order to submit partial results to a history information request

The ELO of the Requested MSA does not have all the requested information available and decides to send back the available information and to announce complementary information later. In that case, the partial results are sent by the Requested MSA application to the Requesting MSA application via a history results message (IE820: C_HIM_RES). The history results message (IE820: C_HIM_RES) indicates that complementary information is still expected with the <Complement of Information Expected flag> Data Item equal to “1”. Finally, the Requested MSA application sets the state at the Requested MSA from “*Open*”, “*Extended*”, “*Late*” or “*Partially answered*” to “*Partially answered*”.

The Requesting MSA application validates the received history results (IE820: C_HIM_RES) structure and checks that the <Complement of Information Expected flag> Data Item is “1”. The Requesting MSA application changes the state of the concerned request from “*Open*”, “*Extended*”, “*Late*” or “*Partially answered*” to “*Partially answered*”. Finally, the Requesting MSA stops the TIM_HIS timer.

The scenario for the submission of partial results to a history information request is depicted in Figure 194 and Figure 195:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

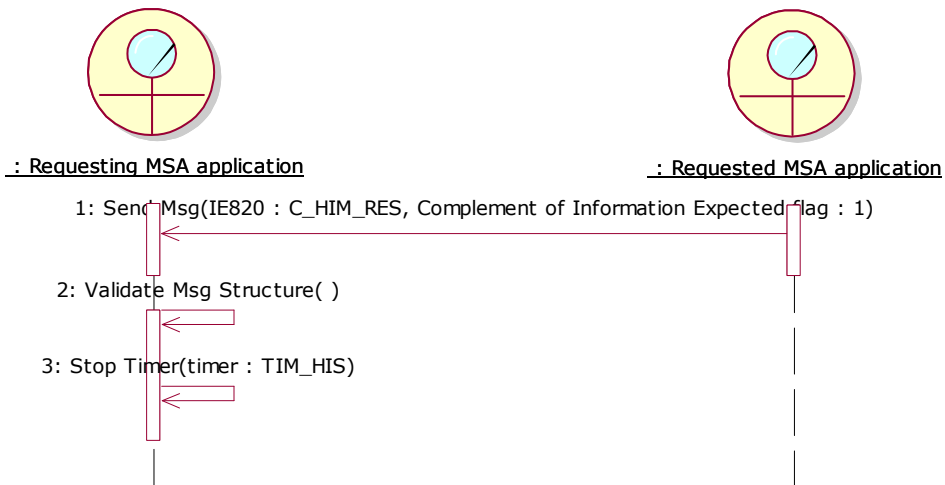


Figure 194: TSD - Submission of partial results to a history information request

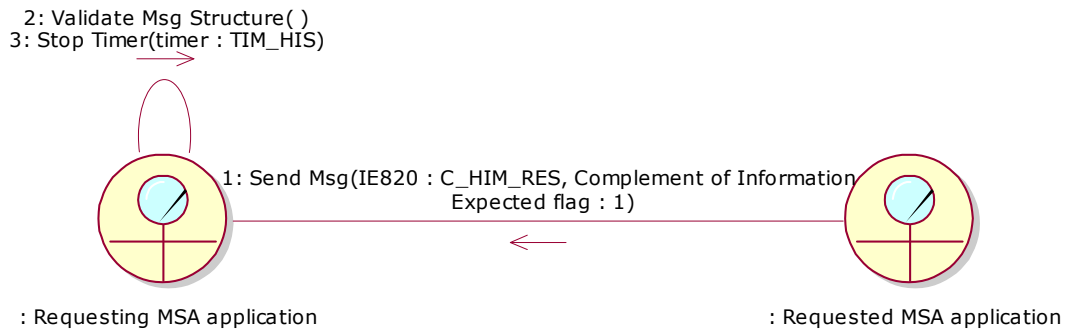


Figure 195: CLD - Submission of partial results to a history information request

IV.III.2 State-Transition Diagrams for Request for history scenarios

IV.III.2.1 Requesting MSA

The following State Transition Diagram in Figure 196 depicts the State Machine at the Requesting MSA. All the transitions are described in detail in Sub-Section IV.II.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

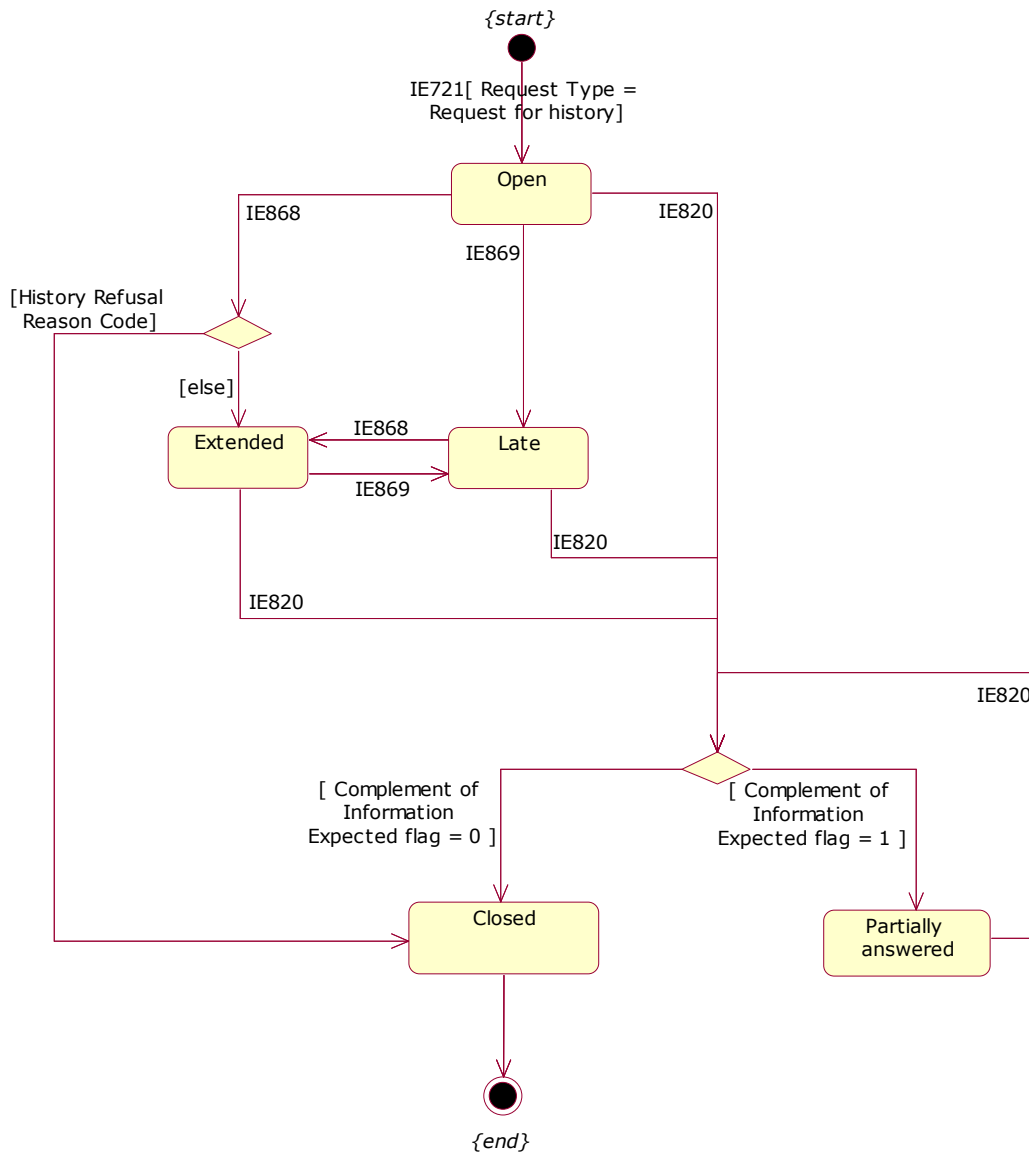


Figure 196: STD at Requesting MSA for Request for history information scenarios

IV.III.2.2 Requested MSA

The following State Transition Diagram in Figure 197 depicts the State Machine at the Requested MSA. All the transitions are described in detail in Sub-Section IV.II.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

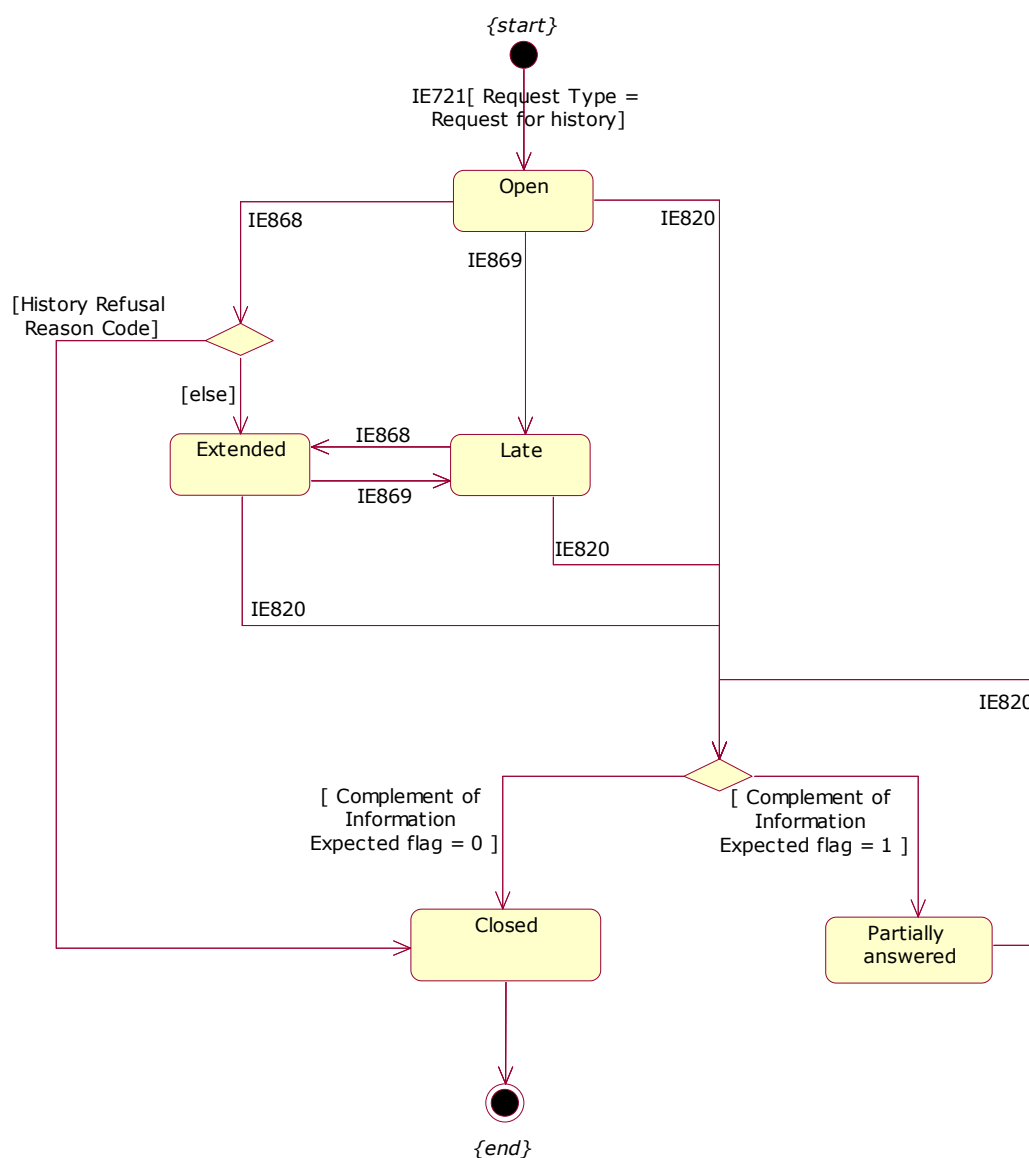


Figure 197: STD at Requested MSA for Request for history information scenarios

Sub-Section IV.IV Movement verification for duty paid movements scenarios

The movement verification is an administrative tool for the verification of movements of excise goods after their release for consumption. The exchange protocol for the movement verification use cases involves two MSAs the requesting and the requested MSAs.

The scenarios of this section describe the “Movement Verification – Request (UC3.15)” and the “Movement Verification – Reminder (UC3.16)” use cases, through the following elementary scenarios:

- Submission of a movement verification request (see IV.IV.1);
- Reminder at expiry time for sending the results to a movement verification request (see IV.IV.2);

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

- Submission of an answer for extending the deadline (see IV.IV.3);
- Submission of the results to a movement verification request (see IV.IV.4).

The movement verification scenarios message exchanges cause state transitions, which are depicted in section IV.IV.5 State-Transition Diagrams for Movement verification for duty-paid movements scenarios.

Concerning the coordination protocol validations, invalid values of “MV Correlation ID” should be refused via an IE906 message (refer to VIII.I.3.2.3.1 Coordination protocol validations).

IV.IV.1 Submission of a movement verification request

This scenario describes the initiation of a movement verification request by the Requesting MSA. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included.

The Requesting MSA application submits a movement verification request (IE722: C_MVS_SUB) to the Requested MSA application. The movement verification request indicates the deadline by which the results of the requested verifications should be received from the Requested MSA. The Requesting MSA application initiates the timer TIM_MVS to expire at the expected deadline for results as specified in the request message. The state of the request at the Requesting MSA is set to “Open”.

The Requested MSA application validates the received message structure and stores the message whilst setting the state to “Open” at the Requested MSA.

The scenario for the submission of a movement verification request is depicted in Figure 198 and Figure 175:

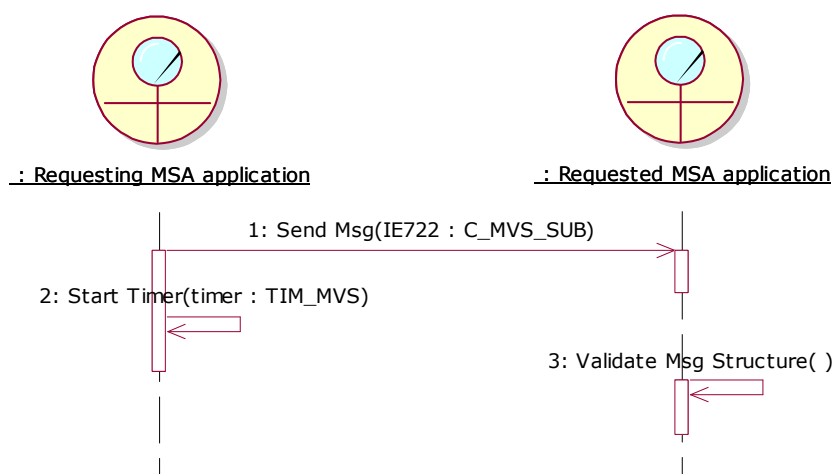


Figure 198: TSD - Submission of a movement verification request

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

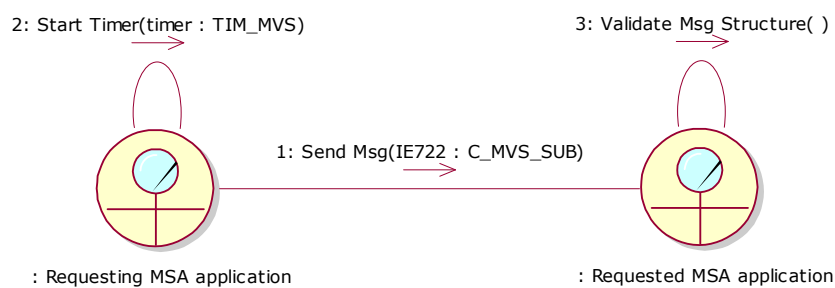


Figure 199: CLD - Submission of a movement verification request

IV.IV.2 Reminder at expiry time for sending the results to a movement verification request

The scenario describes the case that the Requested MSA has not received the results to a movement verification request within time and the timer TIM_MVS expires. The scenario requires that the concerned request is at “Open” or “Extended” state.

Section	Scenario	Ending State
IV.IV.1	Submission of a movement verification request	Open
IV.IV.3	Submission of an answer for extending the deadline	Extended

Table 26: Scenarios that must have been previously completed in order to send a reminder for a movement verification request

At the Requesting MSA application, the timer TIM_MVS expires without having received neither an answer message (IE723: C_MVS_ANS) nor a results message (IE725: C_MVS_RES) for extending the deadline from the Requested MSA application. Alternatively, an answer message (IE723: C_MVS_ANS) had been received from the Requested MSA application but the timer TIM_MVS, that was reset following the reception of the answer message, expires without having received the results message (IE725: C_MVS_RES). The Requesting MSA application then sends a reminder message for a movement verification request (IE724: C_MVS_REM) to the Requested MSA application. The Requesting MSA application changes the state of the request from “Open” or “Extended” to “Late”.

The Requested MSA application validates the received message structure and stores the message whilst setting the state from “Open” or “Extended” to “Late” at the Requested MSA.

The scenario for the expiry of the TIM_MVS timer is depicted in Figure 200 and Figure 201:

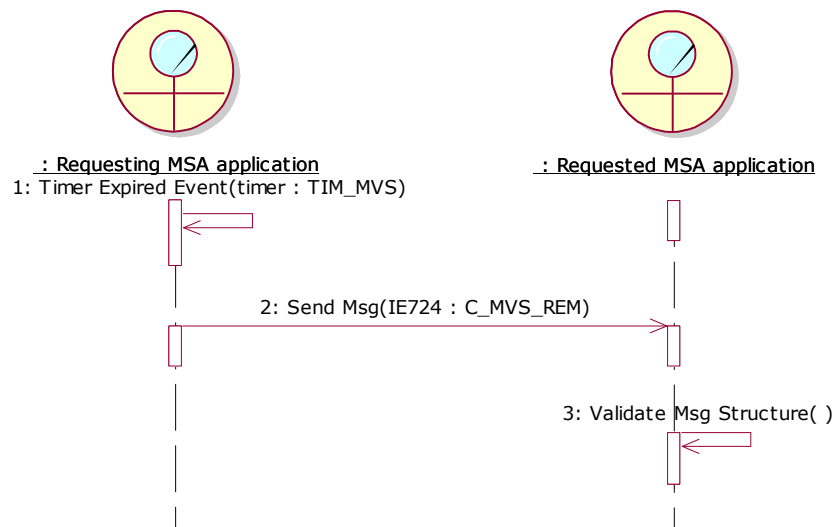


Figure 200: TSD - Reminder at expiry time for sending the results to a movement verification request

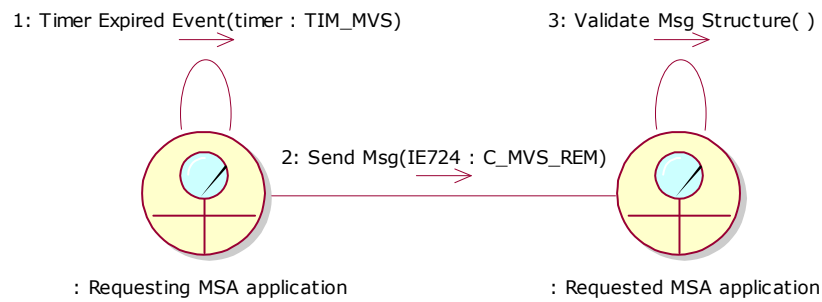


Figure 201: CLD - Reminder at expiry time for sending the results to a movement verification request

IV.IV.3 Submission of an answer for extending the deadline

This scenario assumes that a movement verification request (IE722: C_MVS_SUB) has been submitted by the Requesting MSA and the Requested MSA does not respond within the deadline for results. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires that the concerned request is at “Open” or “Late” state.

Section	Scenario	Ending State
IV.IV.1	Submission of a movement verification request	Open
IV.IV.2	Reminder at expiry time for sending the results to a movement verification request	Late

Table 27: Scenarios that must have been previously completed in order to submit an answer to a movement verification request

The ELO of the Requested MSA examines the contents of the movement verification request and decides to extend the deadline in order to complete the requested verification actions for a given reason. The Requested MSA application sends an answer message (IE723: C_MVS_ANS) to the Requesting MSA extending the deadline for giving the results to the movement verification request. The state of the request at the Requested MSA is set from “Open” or “Late” to “Extended”.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

The Requesting MSA application validates the received message structure and stores the message whilst setting the state of the request from the “Open” or “Late” to “Extended” at the Requested MSA. Then, it resets the timer TIM_MVS to expire at the “Date for which Reply is Requested” date as specified by the answer message (IE723: C_MVS_ANS). In addition, if the timer TIM_MVS has expired, the Requesting MSA application clears the flag that has been raised locally

The scenario for extending the deadline to a movement verification request is depicted in Figure 202 and Figure 203:

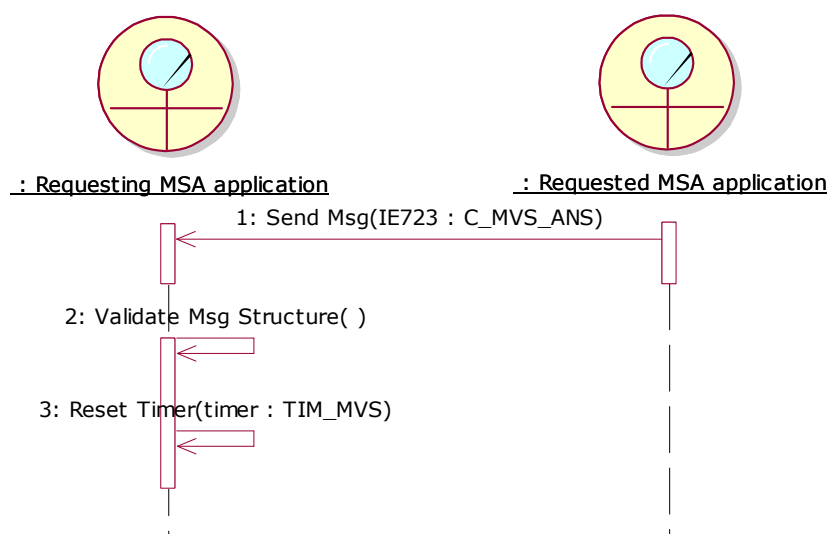


Figure 202: TSD - Submission of an answer for extending the deadline

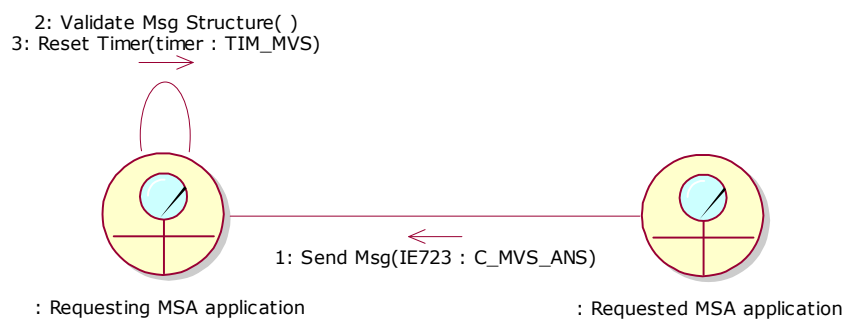


Figure 203: CLD - Submission of an answer for extending the deadline

IV.IV.4 Submission of the results to a movement verification request

The scenario describes the case that the Requested MSA sends the results to a movement verification request. It is assumed that the messages exchanged by the applications are triggered after human intervention, which is not included. The scenario requires the concerned request to be in the “Open”, “Extended” or “Late” state.

Section	Scenario	Ending State
IV.IV.1	Submission of a movement verification request	Open
IV.IV.2	Reminder at expiry time for sending the results to a movement verification request	Late

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.IV.3	Submission of an answer for extending the deadline	Extended
---------	--	----------

Table 28: Scenarios that must have been previously completed in order to submit the results to a movement verification request

The ELO of the Requested MSA completes the requested verification actions specified by the movement verification request (IE722: C_MVS_SUB) from the requesting MSA. The results of these actions (IE725: C_MVS_RES) are then sent by the Requested MSA application to the Requesting MSA application. The Requested MSA application sets the state at the Requested MSA from “Open”, “Late” or “Extended” to “Closed”.

The Requesting MSA application validates the received movement verification results (IE725: C_MVS_RES) message structure and changes the state of the concerned request from “Open”, “Late” or “Extended” to “Closed”. If there is a timer TIM_MVS running the Requesting MSA stops it.

The scenario for sending the results to a movement verification request is depicted in Figure 204 and Figure 205:

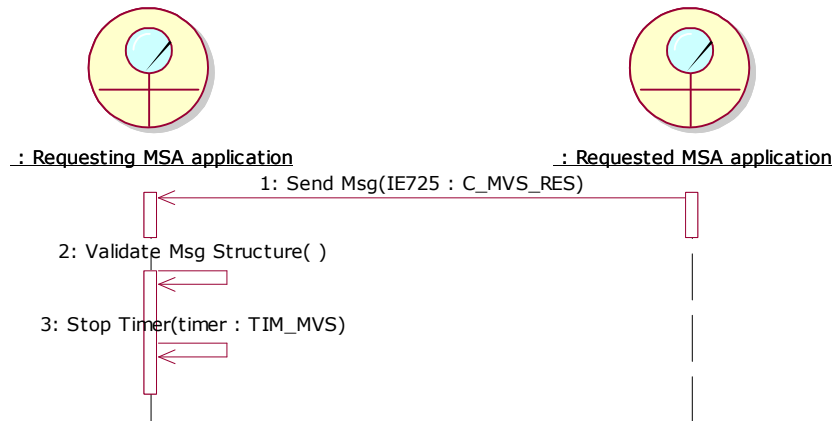


Figure 204: TSD - Submission of the results to a movement verification request

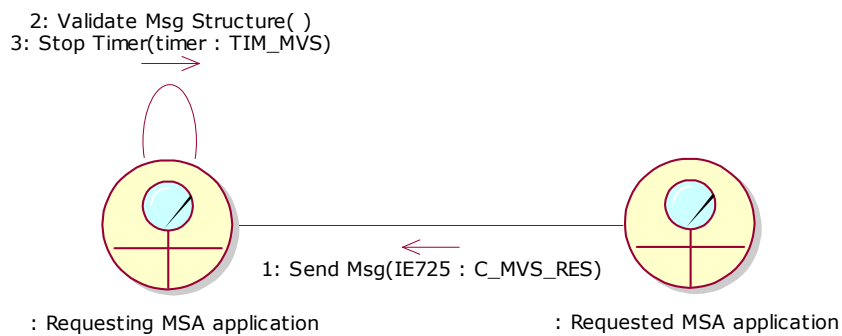


Figure 205: CLD - Submission of the results to a movement verification request

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IV - Follow-up and collaboration	

IV.IV.5 State-Transition Diagrams for Movement verification for duty-paid movements scenarios

IV.IV.5.1 Requesting MSA

The following State Transition Diagram in Figure 206 depicts the State Machine at the Requesting MSA. All the transitions are described in detail in Sub-Section IV.IV.

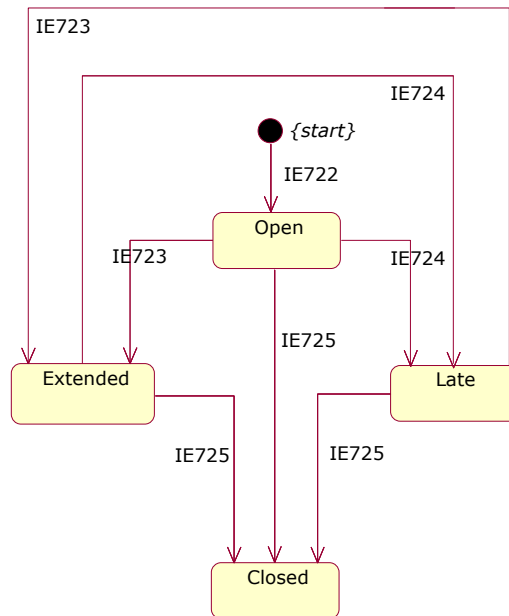


Figure 206: STD at Requesting MSA for Movement verification for duty paid movements scenarios

IV.IV.5.2 Requested MSA

The following State Transition Diagram in Figure 207 depicts the State Machine at the Requested MSA. All the transitions are described in detail in Sub-Section IV.IV.

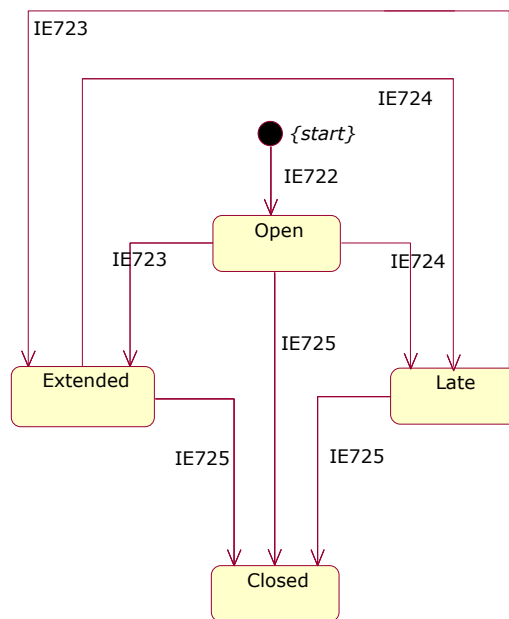


Figure 207: STD at Requested MSA for Movement verification for duty paid movements scenarios

IV.IV.6 Functional Timers

TIM_MVS	
Started:	<p>UC-315-110 – Prepare request message</p> <p>When the application of the Requesting MSA sends a movement verification request (IE722:C_MVS_SUB), it starts the TIM_MVS timer to expire at the expected deadline for receiving back the results.</p>
Stopped:	<p>UC-315-130 - Receive results message</p> <p>When the application of the Requesting MSA receives the movement verification results (IE725:C_MVS_RES) it stops the timer TIM_MVS.</p>
Reset:	<p>UC-315-130 Receive results message</p> <p>When the application of the Requesting MSA receives the movement verification results (IE725:C_MVS_RES) and the TIM_MVS timer has already expired, it resets the flag.</p>
Update:	<p>UC-315-120 - Receive answer message</p> <p>When the application of the Requesting MSA receives a movement verification answer (IE723:C_MVS_ANS) it updates the TIM_MVS timer to expire at the new deadline provided in the deadline report.</p>

Table 29: TIM_MVS functional timer

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

Section V CENTRAL SERVICES

This section describes the message exchange protocols between the NDEAs and the Common Domain Central Services applications namely the System for Exchange of Excise Data (SEED) and the Central Services/Management Information System for Excise (CS/MISE).

Sub-Section V.I SEED

The System for Exchange of Excise Data (SEED) is located in the Common Domain. It provides management and dissemination services regarding information on the Economic Operators register and the Reference Data register.

SEED has a two-fold role:

- to comply with Article 22 of Council Regulation (EC) No 2073/2004 of 16th November 2004 on administrative cooperation in the field of excise duties;
- to provide each MSA with an up-to-date copy of the characteristics of all authorised Economic Operators, so that the validation of an e-AD (or of any other data set submitted in the course of EMCS movements) may be completed in a Member State, without having to cross-consult information from MSA to MSA; and
- to provide each MSA with an up-to-date copy of the EMCS Reference Data maintained by the Commission.

This is a vital part of the EMCS Central Services and an important dependency for the EMCS core Business Processes.

V.I.1 Messages involved

The information that is going to be exchanged regarding SEED is grouped in the following categories:

- Reference Data;
- SEED data;
- SEED statistics.

The Information Exchanges planned are:

- Incremental update or full register of Economic Operators C_QRO_DAT (IE713), as identified in the process thread UC-114-110, UC-114-240 and UC-116-210 [A1].
- Refusal of update of Economic Operators C_QRO_REF (IE714), as identified in the process thread UC-114-210 [A1].
- Common request C_REQ_SUB (IE701), as identified in the process thread UC-116-120 and UC-105-110 [A1].
- Refusal of common request C_REQ_REF (IE702).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- Reference Data dissemination C_RDD_DAT (IE734), as identified in the process thread UC-106-110 and UC-105-210 [A1].
- SEED statistics C_STA_VAL (IE742), as identified in the process thread UC-316-210 and UC-316-120 [A1].
- The XML NACK C_XML_NCK (IE917) is used for reporting XML formatting errors.

V.I.2 Exchange of Reference Data

V.I.2.1 Introduction

The *Reference Data* includes:

- a set of common system parameters, i.e. durations, numbers and codes that are set as committing limits for all Member States in some business cases, typically, a maximum time limit that a MSA is forbidden to exceed but is allowed to shorten, or the list of categories of goods allowed at splitting;
- the various lists of codes to be used in identified fields of the information exchanges throughout EMCS; examples are country codes, transport codes and language codes.

Changes of Reference Data are decided by the Commission and applied to SEED by the Common Domain Data Administrator. Changes to the EMCS Reference Data are disseminated by SEED to each MSA according to their dissemination profile preferences. Each MSA is committed to activate the EMCS Reference Data changes at the latest at the date where it becomes applicable.

V.I.2.1.1 Dissemination of Reference Data (UC1.06)

Due to the fact that the Reference Data are of great importance to the correct functioning of EMCS, the Reference Data should be accurate and up-to-date. Part of this responsibility is taken by the Common Domain Central Services. The Common Domain Central Services have as a major responsibility the updating of all Reference Data and the dissemination of these data to the MSAs. The SEED will distribute the Reference Data changes to all MSAs at a given number of days before they become applicable (typically three or four opening days) according to the validity dates stated in the records. These Reference Data changes will be received by each MSA whose responsibility is to apply them as soon as possible and at the latest by the date and time where they become applicable.

In detail, SEED will prepare the update message IE734: C_RDD_DAT that combines the updates of Reference Data to be applied by the MSAs.

The update message IE734 prepared by the SEED will be sent to all the MSAs.

Upon receipt of the update message each MSA applies the updates included in the IE734 message to their database.

The information exchange needed during the dissemination process of Reference Data is depicted in Figure 208:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

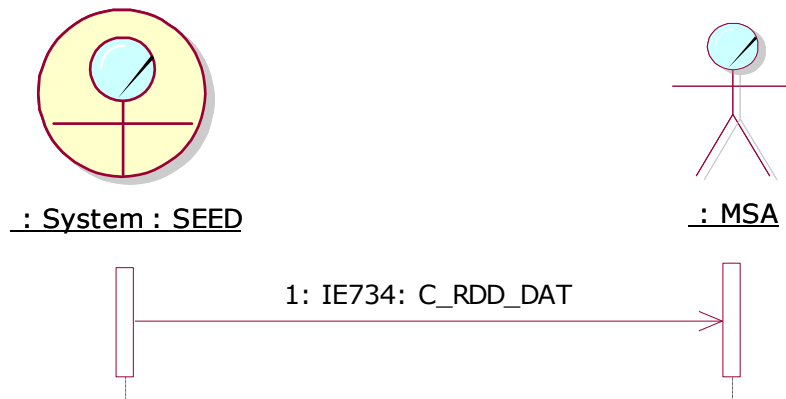


Figure 208: Dissemination of Reference Data

V.I.2.1.2 Re-synchronisation of Reference Data (UC1.05)

In situations where a MSA believes that the Reference Data registered in the system may be corrupted or outdated compared to the one kept by the SEED, a re-synchronisation process of Reference Data is deemed necessary. Re-synchronisation of Reference Data is based on the principle that when a MSA identifies that its local database containing the Reference Data is outdated compared to the central database of the Reference Data in the SEED, the concerned MSA may request the SEED for all or part of the update information in order to obtain an up-to-date Reference Data database.

Specifically, the MSA concerned with the inconsistency of its Reference Data prepares a request message IE701: C_REQ_SUB. The IE701 message shall always specify a date range, an operation value ('Retrieve' or 'Extraction') signifying whether a retrieval or extraction shall be performed and optionally the requested list of codes. However, if no requested list of codes is specified, all EMCS Reference Data types defined by C_RDD_DAT (IE734) will be included in the response message (IE734).

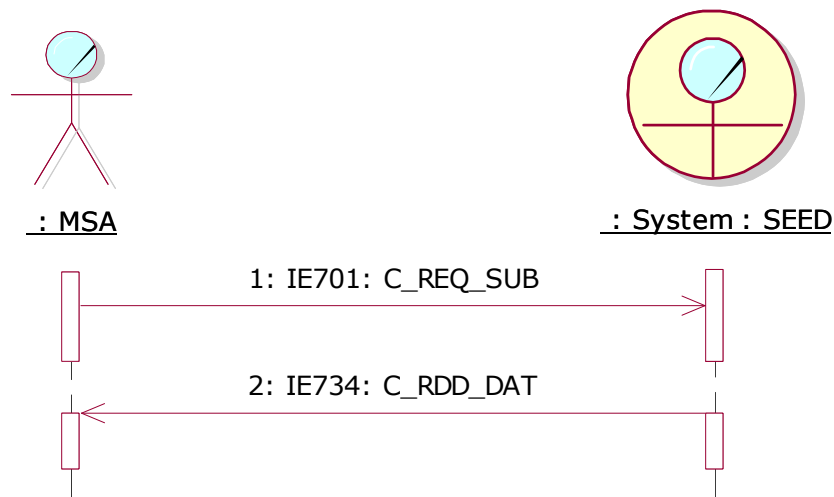


Figure 209: Re-synchronisation of Reference Data

Upon reception of the request message IE701, SEED will send a response message in the form of the IE734: C_RDD_DAT to the requesting MSA. The IE734 will contain the modifications of Reference Data for the period specified in the corresponding request message (IE701).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.I.2.2 Functional ways to get data

V.I.2.2.1 Overview

The functionally different ways for a user to get the data from the repository are:

- *Retrieval* gets a set of modifications that have been applied to selected data types during a given time period (specified by begin and end date). Please note that the increments that are mentioned in the previous paragraph are constituted by such modifications (retrievals).
- *Extraction* gets a set of Data Items, from selected data types, that are valid for a given time period (specified by begin and end date) or a part of that period.
- *Acquisition* of IE734 messages produced automatically and broadcasted by the SEED, every time the Common RD is modified as is specified in the paragraph V.I.2.1.1 Dissemination of Reference Data (UC1.06).
- *Queries* get the set of Common Reference Data entries that match one or more criteria given by the user.

V.I.2.2.2 Retrieval

Retrieval allows the users to retrieve modifications made to the repository during a given time period. These modifications are returned as IE734: C_RDD_DAT messages.

For instance, if a user requests a retrieval of the Common RD data with a time range from the 10th September to the 15th September, he/she will in return receive the list of all the modifications to the Common RD entered between the 10th and the 15th of September.

It should be stressed that *retrieval* functionally is very different from *extraction* because the time range given for the retrieval does not relate to validity dates but to actual dates where the modifications were entered in the repository. For instance, it does not make any sense to perform retrieval for dates in the future.

A full retrieval means retrieving all the modifications to the Common RD since the database was set-up for the first time.

V.I.2.2.3 Extraction

Extraction allows the users to extract the Common RD valid for a given period. This data is returned as IE734: C_RDD_DAT messages.

For instance, if a user requests an extraction of the Common RD for the period from the 10th to the 15th of September and a Reference Data entity is modified with a validity date of 13th of September, the extraction will provide the following information:

- It will give the value of the specific data entity, such as it is valid on the 10th September;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- It will also provide the modification that takes/took effect on the 13th September.

V.I.2.2.4 Acquisition

Acquisition denotes the reception by a MSA of an IE734, which is disseminated automatically by the SEED. The specific functionality is explicitly described in paragraph V.I.2.1.1 Dissemination of Reference Data (UC1.06).

V.I.2.2.5 Queries

Queries allow the user to specify one or more criteria and receive as a result the set of data entries matching these criteria.

The criteria that can be used are:

- Data type;
- Date of validity;
- Value of one of the fields pertaining to a selected data type.

This functionality is covered by the interactive mode of access to SEED.

V.I.2.3 Modes of access to SEED

V.I.2.3.1 CCN/CSI Queue Based Mode

The exchange of IE734, IE732, IE710, and IE709 through CCN/CSI in queue-based mode shall be allowed during operations of EMCS in Phase 3. The protocol for queue-based exchanges of these Information Exchanges on CCN/CSI foresees the basic exchanges of these messages plus error reporting on them. Specifically, the error reports-messages generated are defined in Section X.I.2.2.1:

In this access mode, the following operations are needed in case of the dissemination of Reference Data by the SEED:

- *IE734 sending.* SEED prepares the update message IE734: C_RDD_DAT that groups the changes of Reference Data to be applied by all MSAs. The IE734 which may include all or one of the following messages: IE732, IE710, and IE709, shall be sent to all NEA of MSAs which are connected to CCN/CSI through the CCN network.

In the case of the resynchronisation of Reference Data using the CCN/CSI queue based mode we have the following message exchanges:

- *IE701 sending.* The MSA concerned with the inconsistency of its Reference Data prepares a request message IE701: C_REQ_SUB. The IE701 message which should always specify a date range and an operation value ('Retrieve' or 'Extraction') signifying whether a retrieval or extraction should be performed will be sent to the SEED through the CCN network.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- *IE734 sending.* As a response to the reception of the IE701, SEED will send the IE734: C_RDD_DAT containing the increment update of the Reference Data needed to the requesting MSA. The IE734 will contain the modifications of Reference Data for the period specified in the corresponding request message (IE701).

V.I.2.3.2 Web Services (HTTP/S)

This chapter specifies the normal flow of messages exchanged between MSA and SEED to implement the information exchanges formatted as SOAP over CCN/HTTP (SOAP/HTTP). This mode of access is used for the communication between a MSA and SEED for the purpose of maintaining common Reference Data.

Resynchronisation of Reference Data is a process that may take a long time to complete. It is then necessary to manage long-lived conversations that span over multiple synchronous SOAP requests as explained in Section XI Transport of Messages via SOAP/HTTP. The SOAP calls are synchronous but the execution of the command is asynchronous.

V.I.2.3.2.1 Re-synchronisation of Reference Data

A scenario of the flow of messages for the re-synchronisation of the local Reference Data of the MSA to the one kept by the SEED via asynchronous SOAP/HTTP is shown below. The conversation process is described after the figure.

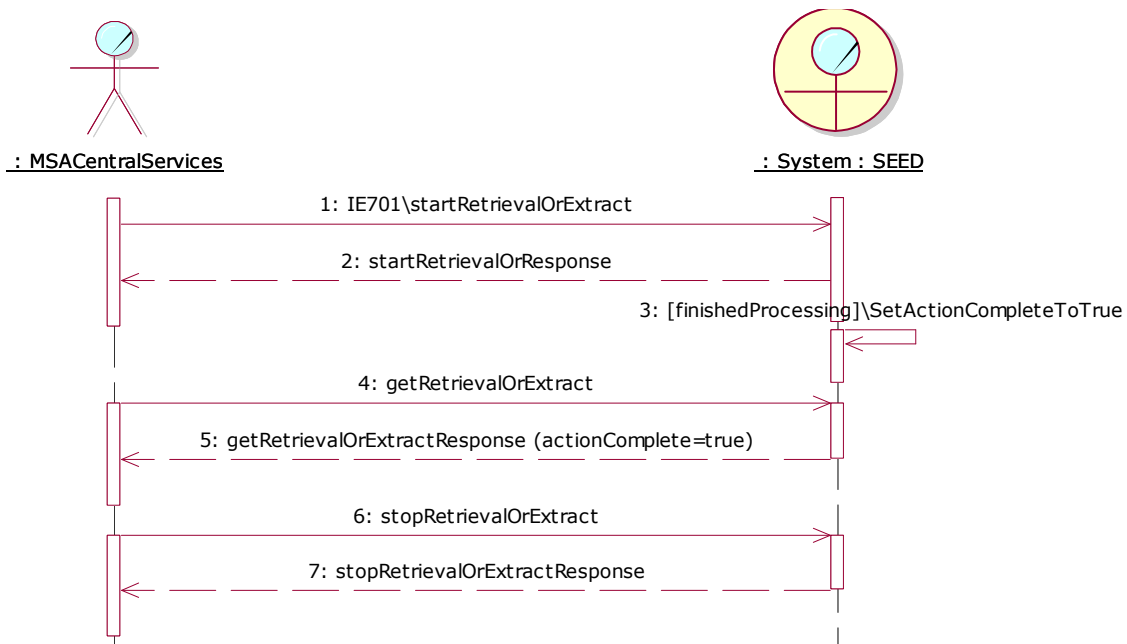


Figure 210: Retrieve or Extract Entity Web Service

In order to submit a request, the MSA may interact with the ‘RetrieveOrExtractEntity’ web service, which exposes an asynchronous interface for this purpose.

A MSA starts an asynchronous conversation with the ‘RetrieveOrExtractEntity’ web service by invoking the ‘startRetrievalOrExtract’ operation with an instance of ‘startRetrievalOrExtract’ entity as parameter (in the body element of the SOAP message) and an IE701 as an attachment.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

For example, inside the body of the SOAP message:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <StartHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <startRetrievalOrExtract
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The MSA will also submit a unique conversationID that will be used for all subsequent requests to identify the requested action. Information regarding the conversationID and the details of the SOAP conversation is provided in Section XI Transport of Messages via SOAP/HTTP. The 'RetrieveOrExtractEntity' web service sends upon reception an empty 'startRetrievalOrResponse' entity.

The web service might need a lot of time to prepare the requested data that should be sent to the MSA. The processing includes syntax and semantics validation.

During the processing time, the MSA can invoke at regular time intervals the 'getRetrievalOrExtract' operation.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getRetrievalOrExtract
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The web service sends a 'getRetrievalOrExtractResponse' entity to the MSA.

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
```


EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

```

<getRetrievalOrExtractResponse
  xmlns="http://emcs.dgtaxud.ec/webservice/types">
  <ActionSucceeded>>false</ActionSucceeded>
</getRetrievalOrExtractResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

- If the 'ActionSucceeded' field value is a "false" Boolean value, the processing is not yet finished and the MSA must retry the call later;
- If the 'ActionSucceeded' field value is a "true" Boolean value, the processing is finished and the MSA must check if the "ActionResult" response field contains the 'Error' string. If there are no errors ('ActionResult' contains 'Success'), the response contains an IE734 as an attachment with instances matching the criteria given in the submitted IE701.

For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <getRetrievalOrExtractResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types">
      <ActionSucceeded>>true</ActionSucceeded>
      <ActionResult>Success</ActionResult>
    </getRetrievalOrExtractResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Finally, the MSA must terminate the conversation with a call to the 'stopRetrievalOrExtract' operation by sending a 'stopRetrievalOrExtract' entity.

For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <stopRetrievalOrExtract
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SEED frees all its internal resources locked by the conversation and returns immediately an empty 'stopRetrievalOrExtractResponse' entity.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <stopRetrievalOrExtractResponse
      xmlns="http://emcs.dgtaxud.ec/webService/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

V.I.2.3.2.1.1 Error Management

All SOAP/HTTP exchanges between a MSA and SEED share the same basic error management. Upon reception of a syntactically or semantically incorrect message (SOAP message), the web service notifies the application of the syntax or semantic error by sending a SOAP Fault message, which details the reason for rejecting the message. This message contains both the reason for the rejection and the location in the original message of the element that caused the error.

It must be stressed that if there is an error (no compliance with the format or rules and conditions of IE701) in the received instance of IE701, then the web service does not send a SOAP Fault message, instead an IE702 or an IE917 is sent in the 'Error' entity returned by a 'getActionStatus' operation.

SOAP Faults are further explained in Section XI Transport of Messages via SOAP/HTTP.

V.I.2.3.3 Interactive mode

The Interactive mode of access will be implemented via accessing the SEED on the web. It shall allow all users to perform "extraction" and/or "retrieval" of data as well as to make queries on the database via the GUI. Moreover, it shall allow users having the appropriate access rights (CD.Data Administrator) to prepare and upload the CD Reference Data modifications via the GUI. The HTTPS protocol is used.

V.I.3 Exchange of SEED data

V.I.3.1 Introduction

V.I.3.1.1 The Role of SEED data

SEED information includes all items that are described in Article 22 of Council Regulation (EC) No 2073/2004, namely:

- authorised warehouse keepers (as defined in Directive 2008/118/EC);
- registered Consignees (as defined in Directive 2008/118/EC);
- other registered operators that a MSA may allow to provide a movement guarantee in place of the Consignor; they may be considered as "persons who have assumed the obligations within the meaning of Article 18(1) of Directive 2008/118/EC";

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- Temporary authorisations (as defined in Directive 2008/118/EC).

It should be stressed that the register of the SEED information will have the following content:

- the identification number issued by the competent authority regarding the person or premises (also known as the "Excise number");
- the name and address of the person or premises;
- the category of excise products which may be held or received by the person or which may be held or received at these premises;
- identification of the central liaison office or the excise office from which further information may be obtained;
- the date of issue, amendment and where applicable, the date of cessation of validity of the authorisation.

In addition to the above-mentioned information which is part of the SEED register, it should be mentioned that other relevant information such as specific authorisations (i.e. the allowance to practise direct delivery or to send energy products without identified destination under Article 22 of Directive 2008/118/EC) is contained in the register as well

Last but not least, temporary authorisations granted by a MSA of Destination to a non-registered Consignee (as defined in Directive 2008/118/EC) are also part of the **SEED information**. A temporary authorisation can cover one or several movements. In both cases, a temporary authorisation can only concern one Consignor and one Consignee for a given period of validity and a certain quantity of goods (expressed in the unit associated with the product code).

It should be mentioned that the main purpose of the usage of the SEED information is the formal validation of the e-AD and of all related submissions.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.I.3.1.2 Dissemination of SEED data (UC1.14)

The maintenance and dissemination of the SEED data is of great importance to the SEED application. The registration information of the Economic Operators must be in all cases accurate and up-to-date, which means that the distribution of changes on these data is critical. A MSA should submit its updates of SEED data a sufficient time in advance from the date where they become legally valid and at the latest on the activation date itself so that they are ready for use in all Member States on that date. In case of urgent updates such as immediate withdrawal of an authorisation, the MSA is requested to submit the update as soon as possible. Thus, each MSA is responsible for providing any changes of its register of Economic Operators to all other MSAs in due time. For consistency purposes, economic operator changes from each MSA are concentrated and consolidated in the common reference version of SEED maintained by the Common Domain Central Services-SEED. The SEED application is responsible for the dissemination of the register updates to all Member States that opted for receiving IE713 disseminations, as soon as they are integrated to the central SEED repository. Specifically, an agent in the MSA Central Services sends the set of update records to SEED application at the Common Domain Central Services (SEED) in the form of an IE713: C_QRO_DAT message.

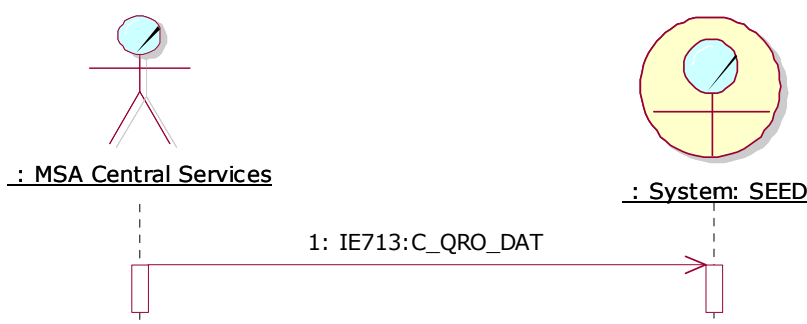


Figure 211: Dissemination of SEED data from MSA Central Services to SEED

After receiving this message, SEED performs a formal validation in connection with pre-existing information.

In particular:

- Validate message against the applicable DDNEA rules and conditions;
- Validate that the entities are created/updated/invalidated only by the owner MSA;
- Validate relationships between dates. The start date of an excise authorisation must precede the date when the authorisation expires. The activation date should be equal to or greater than the modification date. The end date of an excise authorisation can be left unspecified ('until further notice').

If the changes are found to be invalid, the SEED application at the Common Domain Central Services will reply with the transmission of an IE714: C_QRO_REF message signifying a refusal of update of Economic Operators and thus the submitted update will be completely rejected. The refusal of update for the economic operator register is depicted in the figure below:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

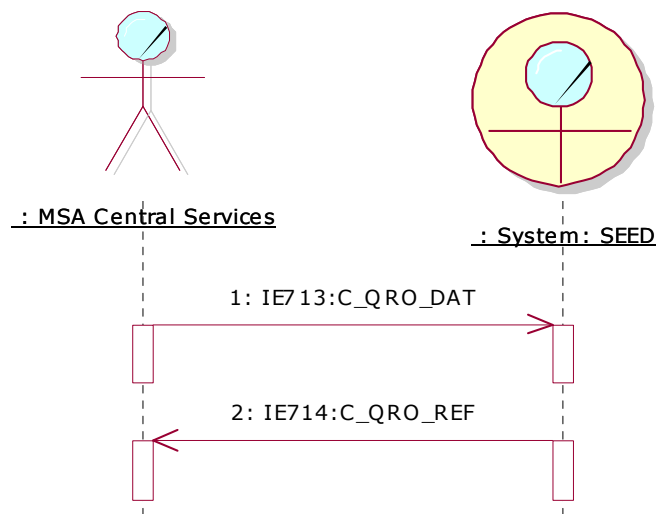


Figure 212: Dissemination of SEED data where refusal of Economic Operators update occurs

On the other hand, if after the validation process the submitted update is accepted, the SEED application will prepare a unique consolidated message (IE713) containing all the validated updates related to the Economic Operators and will send them back to all Member States that opted for receiving IE713 disseminations, as soon as they are integrated to the central SEED repository.

The unique consolidated update message is composed of all updates submitted by the MSAs and validated by the Common Domain Central Services, under the form of actions such as Create, Update or Invalidate from the preceding state of the register. All Member States are committed to use the state of information received back from the Common Domain to update their National SEED register. The NDEA is responsible for applying the consolidated updates (IE713: C_QRO_DAT) as soon as they are received from SEED. The Member States using the SOAP channel should poll Central SEED for updates at least twice a day.

It shall be noted that a modification with an activation date set to today becomes applicable immediately. That is, from the time the modification was accepted by central SEED and disseminated to all MSAs (recorded by the “Modification Date and Time” data item). Planned modifications having an activation date in the future become applicable at that date. The validation of a message at the MSA of Dispatch is done against the current SEED data (that is, the data that exist in SEED at the time of validation).

MSAs sending updates to central SEED should wait until they get a response from central SEED, either in the form of dissemination (IE713) or in the form of refusal (IE714) of the proposed updates, before they initiate a movement involving any of the updated registration data.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

The number of entities contained in the consolidated updates (IE713: C_QRO_DAT) that each NDEA receives is configurable via the SEED web interface. Each MSA may specify a number between 1,000 and 10,000 as the upper limit of entities to be contained in the consolidated messages (IE713) sent by SEED. If a consolidated message (IE713) to be disseminated exceeds the upper limit set by a MSA, SEED will split the consolidated updates and send multiple IE713: C_QRO_DAT messages so that no message exceeds the upper limit while ensuring that all updates are sent. If a MSA wishes to receive consolidated messages (IE713) without an upper limit, it may access the SEED web interface and set the upper limit to 'null'. More information concerning the maintenance of SEED dissemination profiles can be found in SEEDv1 SRD [R11].

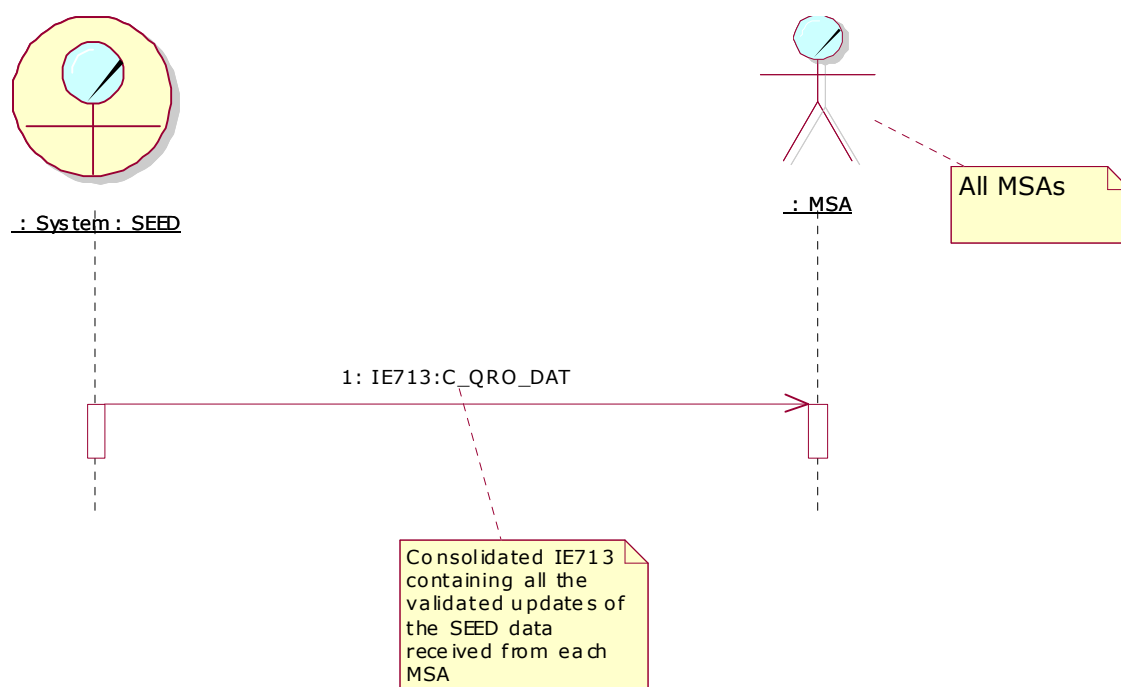


Figure 213: Dissemination of SEED data where update of Economic Operators is accepted

V.I.3.1.3 Re-synchronisation of SEED data (UC1.16)

There may be cases where the SEED database of the MSAs is found to be unsynchronised with the reference version maintained by the Common Domain. In such a case the re-synchronisation of SEED data is required, and the concerned MSA may request the Common Domain Central Services for all or part of the update information in order to obtain an up-to-date SEED database. This is performed by sending a request message IE701: C_REQ_SUB which should always specify a date range and an operation value ('Retrieve' or 'Extraction') signifying whether a retrieval or extraction should be performed. Upon reception of the IE701 message, the SEED application maintained by the Common Domain Central Services, will construct the IE713: C_QRO_DAT message which will contain the updates of the SEED data requested by the concerned MSA.

The IE713 message constructed by the SEED application in the Common Domain Central Services will be sent to the concerned MSA.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

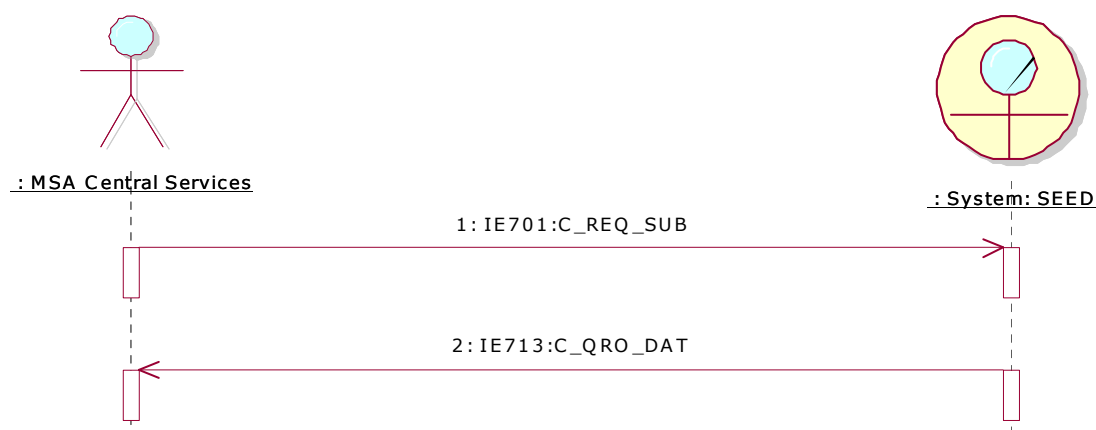


Figure 214: Re-synchronisation of SEED data

V.I.3.2 Functional ways to get data

V.I.3.2.1 Overview

The functionally different ways for a user to get the data from the SEED repository are:

- *Retrieval* gets a set of modifications that have been applied to selected data types during a given time period (specified by begin and end date). Please note that the increments that are mentioned in the previous paragraph are constituted by such modifications (retrievals).
- *Extraction* gets a set of Data Items, from selected data types, that are valid for a given time period (specified by begin and end date) or a part of that period.
- *Acquisition* of IE713 messages produced automatically and broadcasted by the SEED, every time the SEED data is modified as is specified in the paragraph V.I.3.1.2 Dissemination of SEED data.
- *Queries* get the set of Common Reference Data entries that match one or more criteria given by the user.

V.I.3.2.2 Retrieval

Retrieval allows the users to retrieve modifications made to the repository during a given time period. These modifications are returned as IE713: C_QRO_DAT messages.

For instance, if a user requests a retrieval of the SEED data with a time range from the 10th September to the 15th September, he/she will in return receive the list of all the modifications to the SEED entered between the 10th September and the 15th September.

It should be stressed that *retrieval* functionally is very different from *extraction* because the time range given for the retrieval does not relate to validity dates but to actual dates where the modifications were entered in the repository. For instance, it does not make any sense to perform retrieval for dates in the future.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

A full retrieval means retrieving all the modifications to the SEED data since the database was set-up for the first time.

V.I.3.2.3 Extraction

Extraction allows the users to extract the SEED data valid for a given period. This data is returned as IE713: C_QRO_DAT messages.

For instance, if a user requests an extraction of the SEED data for the period from the 10th September to the 15th September and a SEED data entity is modified with a validity date of 13th September, the extraction will provide the following information:

- It will give the value of the specific data entity, such as it is valid on the 10th September;
- It will also provide the modification that takes/took effect on the 13th September.

V.I.3.2.4 Acquisition

Acquisition denotes the reception by a MSA of an IE713: C_QRO_DAT sent automatically by the system as is specified in the paragraph V.I.3.1.2 Dissemination of SEED data (UC1.14).

V.I.3.2.5 Queries

Queries allow the user to specify one or more criteria and receive as a result the set of data entries matching these criteria.

Some indicative criteria can be the following:

- the excise authorisation;
- Authorisation Begin Date;
- Authorisation End Date;
- Operator type, etc.

This functionality is covered by the interactive mode of access to SEED.

V.I.3.3 Modes of access to SEED

V.I.3.3.1 CCN/CSI Queue Based Mode

The exchange of IE713, IE714 and IE701 through CCN/CSI in queue-based mode shall be allowed during operations of EMCS in Phase 3. The protocol for queue-based exchanges of these Information Exchanges on CCN/CSI foresees the basic exchanges of these messages plus error reporting on them. Specifically, the error reports-messages generated are defined in Section X.I.2.2.1.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

In this access mode, the following operations are needed in case of the dissemination of SEED data by the SEED system:

- *IE713 sending.* An agent in the MSA Central Services sends the set of update records to the SEED application at the Common Domain Central Services in the form of an IE713: C_QRO_DAT message through the CCN network.

The received IE713 message will be put under formal validation in comparison to pre-existing information.

- *IE714 sending.* In the case that an IE713 submitted from a MSA is found to be invalid, the SEED application will completely reject the submitted change and produce a refusal of update of Economic Operators message IE714 that will be transmitted to the NEA of the concerned MSA connected to CCN/CSI through the CCN network.
- *IE713 sending.* Upon acceptance of the validated IE713 sent by the concerned MSA, the SEED application inserts the validated updates into the increment of the register of Economic Operators currently under preparation. The increment is composed of all validated updates submitted by the MSAs. Based on a predefined size and a maximum time elapsed between two updates, the consolidated incremental update IE713 is disseminated to the NEA of all MSAs through the CCN network.

In the case of the resynchronisation of SEED data using the CCN/CSI queue based mode we have the below message exchanges:

- *IE701 sending.* The MSA concerned with the consistency of its SEED data prepares a request message IE701: C_REQ_SUB. The IE701 message which should always specify a date range and an operation value ('Retrieve' or 'Extraction') signifying whether a retrieval or extraction should be performed will be sent to the Common Domain Central Services (SEED) through the CCN network.
- *IE713 sending.* As a response to the reception of the IE701, SEED will send the IE713: C_QRO_DAT containing the increment update of the SEED data needed by the requesting MSA.

V.I.3.3.2 Web Services (HTTP/S)

This chapter specifies the normal flow of messages exchanged between MSA and SEED to implement the information exchanges formatted as SOAP over CCN/HTTP (SOAP/HTTP). This mode of access is used for the communication between a MSA and SEED with the purpose of maintaining SEED data.

Dissemination and Resynchronisation of SEED data are processes that may take a long time to be completed. It is then necessary to manage long-lived conversations that span over multiple synchronous SOAP requests as explained in Section XI Transport of Messages via SOAP/HTTP. The SOAP calls are synchronous but the execution of the command is asynchronous.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.I.3.3.2.1 Dissemination of SEED data (Reception of Updates from the MSAs)

A scenario of the flow of messages for the disseminating SEED data from a MSA to the Common Domain SEED via asynchronous SOAP/HTTP is shown below (in essence only the reception of the incremental updates sent by the MSAs is depicted, since the actual dissemination only occurs through CCN/CSI communication channel). The conversation process is described after the figure:

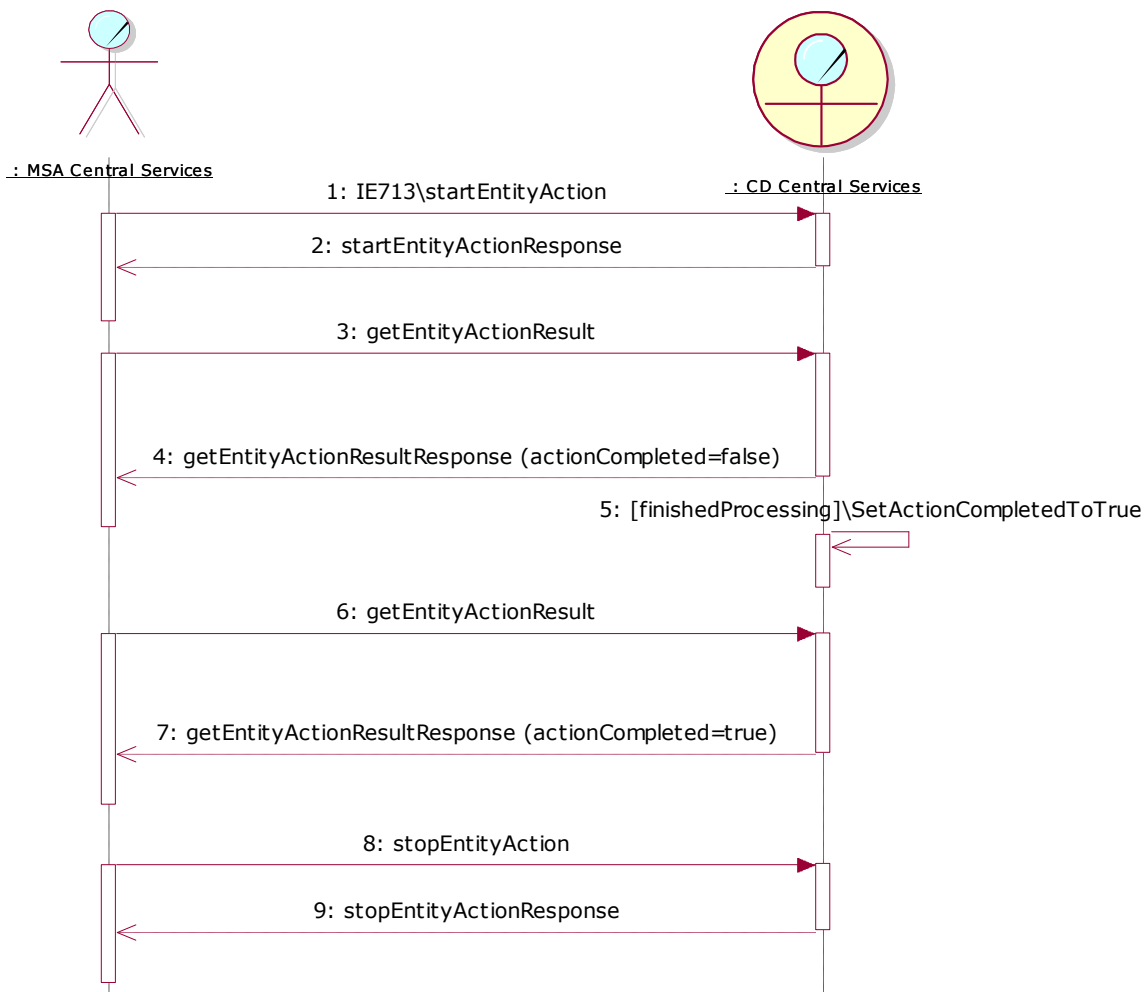


Figure 215: Maintain Entity Web Service

In order to submit update information regarding the Economic Operators, the MSA may interact with the ‘MaintainEntity’ web service that exposes an asynchronous interface for this purpose.

A MSA starts an asynchronous conversation with the ‘MaintainEntity’ web service by invoking the ‘startEntityAction’ operation with an instance of ‘startEntityAction’ entity as parameter (in the body element of the SOAP message). The request contains an instance of an IE713 (C_CRO_DAT) as an attachment.

For example, the SOAP message should be structure like:

```

<?xml version="1.0" encoding="UTF-8" ?>
  
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

```

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <StartHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <startEntityAction
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The MSA will also submit a unique conversationID that will be used for all subsequent requests to identify the requested action. Information regarding the conversationID and the details of the SOAP conversation is provided in Section XI Transport of Messages via SOAP/HTTP. The 'MaintainEntity' web service sends upon reception an empty 'startActionResponse' entity.

The web service might need a lot of time to process the validity of an IE713 message. The processing includes syntax and semantics validation and validation of the submitted information in connection with pre-existing information in SEED.

During the processing time, the MSA can invoke at regular time intervals the 'getEntityActionResult' operation by sending a 'getEntityActionResult' entity. For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getEntityActionResult
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The web service checks if the action has been completed and sends accordingly a 'getActionStatusResponse' entity to the MSA. For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getEntityActionResultResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types">
      <ActionSucceeded>>false</ActionSucceeded>
    </getEntityActionResultResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

```

    </getEntityActionResultResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

- If the 'ActionSucceeded' field value is a "false" Boolean value, the processing is not yet finished and the MSA must retry the call later;
- If the 'ActionSucceeded' field value is a "true" Boolean value, the processing is finished and the MSA must check if the response contains an 'Error' entity of type 'EntityType'. If there are no errors, the processing of all the actions in IE713 was successful.

Finally, the MSA must terminate the conversation with a call to the 'stopEntityAction' operation by sending a 'stopEntityAction' entity. For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <stopEntityAction
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SEED frees all its internal resources locked by the conversation and returns immediately an empty 'stopEntityActionResponse' entity. For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <stopEntityActionResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The submitted instance of IE713 (if the validation has been completed successfully) will be consolidated in an incremental update (probably along with other submitted instances of IE713) and sent at some later time by the Common Domain Central Services to all MSAs using the CCN/CSI or the CCN Mail 2.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.I.3.3.2.1.1 Rejection of Update

In addition to standard error management, SEED may be unable to process the asynchronous request for update because the modifications might violate the data integrity of the central repository. The normal flow of messages for rejection via asynchronous SOAP/HTTP is shown below:

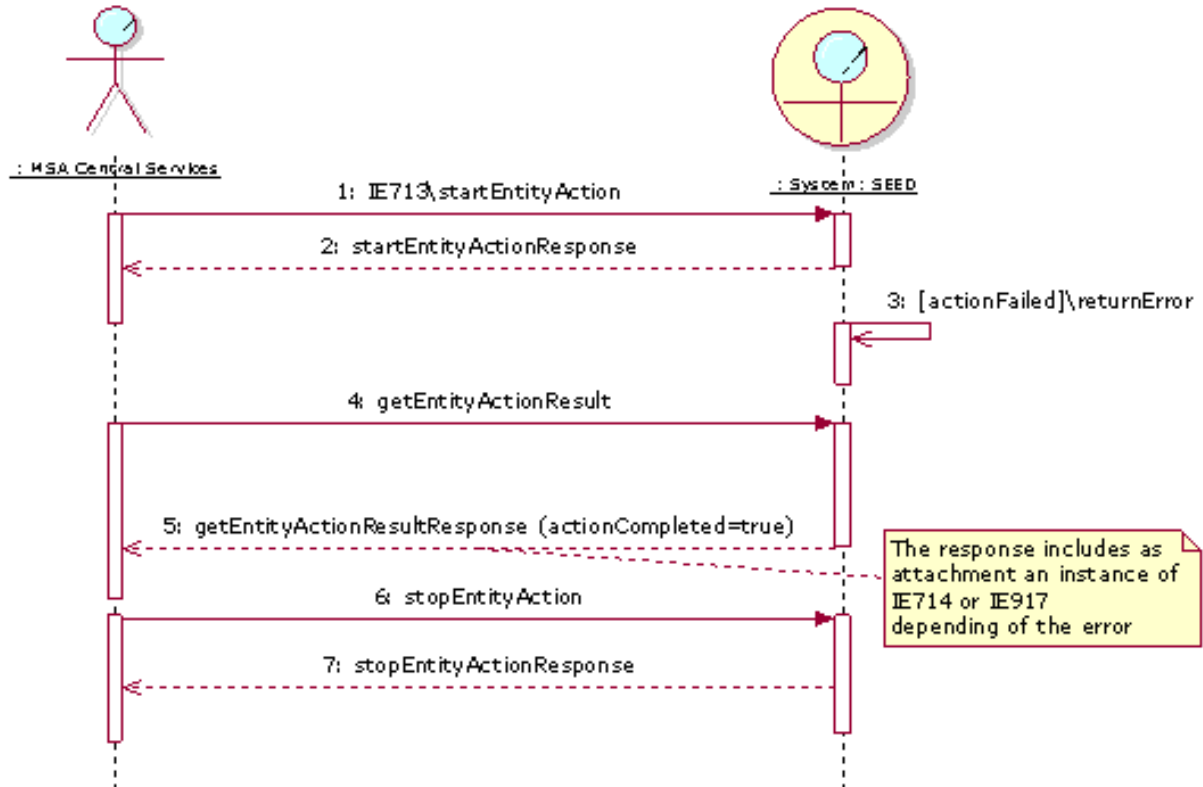


Figure 216: Maintain Entity Web Service (Rejection of Updates)

The main difference from the scenario where the submitted update (IE713) has been accepted is the response of the 'getEntityActionResult' operation. When the web service finds an error during the validation of the submitted update sends an instance of IE714 or IE917 as an attachment to the response message.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <getEntityActionResultResponse
    xmlns="http://emcs.dgtaxud.ec/webService/types">
    <ActionSucceeded>true</ActionSucceeded>
    <ActionResult>Error</ActionResult>
  </getEntityActionResultResponse >
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The IE714 and IE917 messages are sent when the submitted instance of IE713 does not comply with the syntax and semantics rules of the IE713.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

Finally, the MSA must terminate the conversation with a call to the 'terminateAction' operation by sending a 'terminateAction' entity.

V.I.3.3.2.1.2 Error Management

All SOAP/HTTP exchanges between a MSA and SEED share the same basic error management. Upon reception of a syntactically or semantically incorrect message (SOAP message), SEED notifies the application of the syntax or semantic error by sending a SOAP Fault message, which details the reason for rejecting the message. This message contains both the reason for the rejection and the location in the original message of the element that caused the error.

It must be stressed that if there is an error (no compliance with its format or rules and conditions of IE713) in the received instance of IE713, then SEED sends an IE714 or an IE917 as described in the previous paragraph and not a SOAP Fault message.

SOAP Faults are further explained in Section XI Transport of Messages via SOAP/HTTP.

V.I.3.3.2.2 Re-synchronisation of SEED data

The normal flow of messages for re-synchronisation of the local registry of the MSA to the one kept by the Common Domain Central Services via asynchronous SOAP/HTTP is shown below:

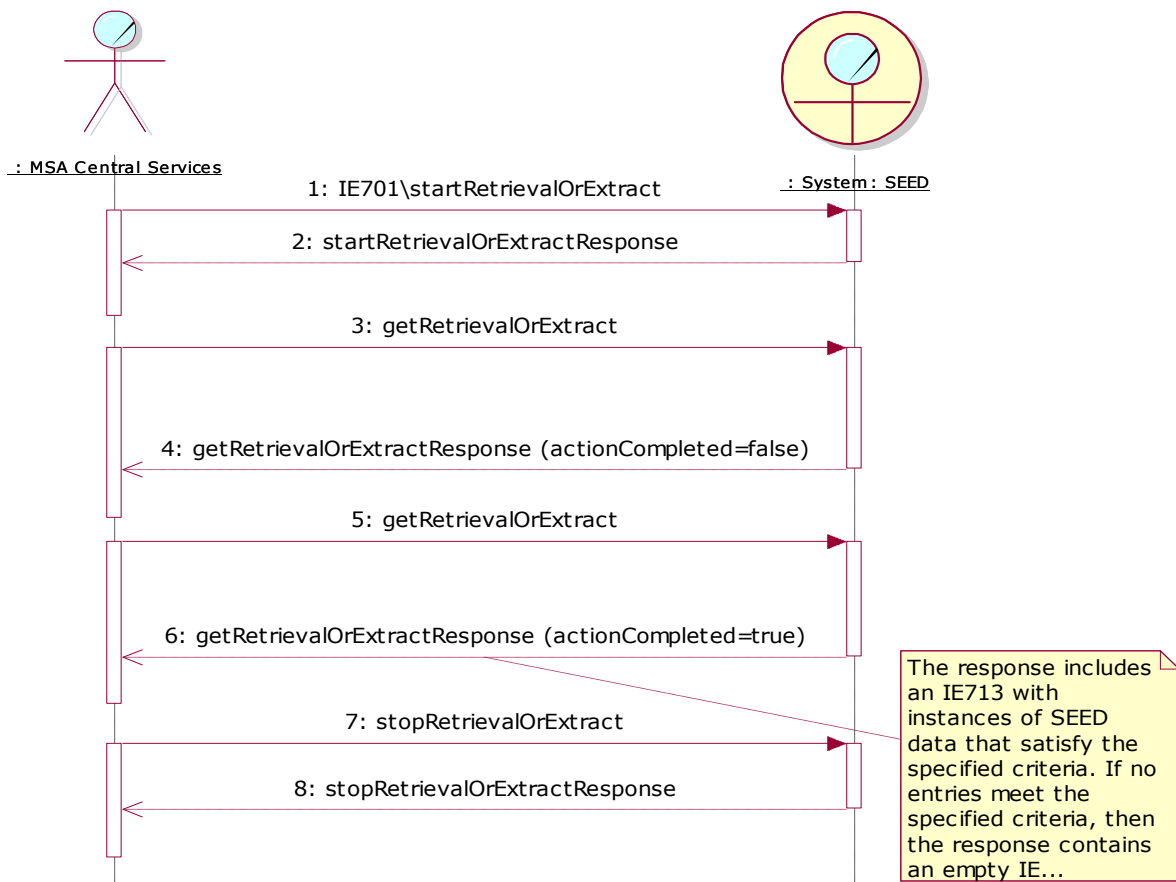


Figure 217: Retrieve or Extract Entity Web Service

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

In order to submit a request update of register of Economic Operators, the MSA may interact with the 'RetrieveOrExtractEntity' web service which exposes an asynchronous interface for this purpose.

A MSA starts an asynchronous conversation with the 'RetrieveOrExtractEntity' web service by invoking the 'startRetrievalOrExtract' operation with an instance of the 'startRetrievalOrExtract' entity as parameter in the body element of the SOAP message and an IE701 (C_REQ_SUB) as attachment.

For example, inside the body of the SOAP message:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <StartHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <startRetrievalOrExtract
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The MSA will also submit a unique conversationID that will be used for all subsequent requests to identify the requested action. Information regarding the conversationID and the details of the SOAP conversation is provided in Section XI Transport of Messages via SOAP/HTTP. The 'RetrieveOrExtractEntity' web service sends upon reception an empty 'startRetrievalOrExtractResponse' entity.

The web service might need a lot of time to prepare the registration data that should be sent to the MSA. The processing first includes syntax and semantics validation.

During the processing time, the MSA can invoke at regular time intervals the 'getRetrievalOrExtract' operation by sending a 'getRetrievalOrExtract' entity.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getRetrievalOrExtract
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The web service sends a ‘getRetrievalOrExtractResponse’ entity to the MSA.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <getRetrievalOrExtractResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types" >
      <ActionSucceeded>false</ActionSucceeded>
    </getRetrievalOrExtractResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- If the ‘ActionSucceeded’ field value is a “false” Boolean value, the processing is not yet finished and the MSA must retry the call later;
- If the ‘ActionSucceeded’ field value is a “true” Boolean value, the processing is finished and the MSA must check if the ‘ActionResult’ field contains the ‘Error’ string. If there are no errors (‘ActionResult’ field contains ‘Success’), the ‘Success’ entity of type ‘EntityActionSuccessType’ contains a sequence of IE713 responses matching the criteria given in the submitted IE701. If there are no matching increments to be sent, an empty instance of IE713 is conveyed as an attachment.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Body>
    <getRetrievalOrExtractResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types" >
      <ActionSucceeded>true</ActionSucceeded>
      <ActionResult>Success</ActionResult>
    </getRetrievalOrExtractResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Finally, the MSA must terminate the conversation with a call to the ‘stopRetrievalOrExtract’ operation by sending a ‘stopRetrievalOrExtract’ entity.

For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/" >
```


EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

```

    <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
  </ContinueHeader>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <stopRetrievalOrExtract
    xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

SEED frees all its internal resources locked by the conversation and returns immediately an empty 'stopRetrievalOrExtractResponse' entity.

For example:

```

<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <stopRetrievalOrExtractResponse
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

V.I.3.3.2.2.1 Error Management

All SOAP/HTTP exchanges between a MSA and Central Services share the same basic error management. Upon reception of a syntactically or semantically incorrect message (SOAP message), the web service notifies the application of the syntax or semantic error by sending a SOAP Fault message, which details the reason for rejecting the message. This message contains both the reason for the rejection and the location in the original message of the element that caused the error.

It must be stressed that if there is an error (no compliance with the format or rules and conditions of IE701) in the received instance of IE701, then the web service does not send a SOAP Fault message, instead an IE702 or an IE917 is sent in the 'Error' entity returned by a 'getRetrievalOrExtract' operation.

SOAP Faults are further explained in Section XI Transport of Messages via SOAP/HTTP.

V.I.3.3.2.2.2 Interactive Mode

The Interactive mode of access will be implemented via accessing the SEED on the web. It shall allow all users to perform "extraction" and/or "retrieval" of SEED data as well as to make queries on the database via the GUI. Moreover, it will allow users with the appropriate access rights (ND Data Administrators) to prepare and upload the SEED Data modifications via a GUI as a fallback procedure. It relies on the HTTPS protocol.

V.I.4 SEED statistics

V.I.4.1 The Role of SEED statistics

SEED statistics should be available to requesting MSAs (See art. 26 of Regulation 2073/2004). The statistical data include the following five types:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- the number of active and inactive economic operators;
- the pending expirations;
- the types of economic operators;
- the number of authorised economic operators by product or product category;
- the number of changes to excise authorisations.

V.I.4.2 SEED statistics (UC3.16)

When an MSA sends to SEED a request statistics, SEED responds with the statistical data requested. The request should contain the list of Member States, the type of statistics and the period for which SEED should provide the statistical data.

Specifically, the MSA requests the statistics from SEED by submitting a IE701: C_REQ_SUB message. Upon reception of the request message IE701, SEED will send a response message in the form of the IE742: C_STA_VAL to the requesting MSA. The C_REQ_SUB will contain the statistical data for the Member States, the type of statistics and the period specified in the corresponding request message (IE701).

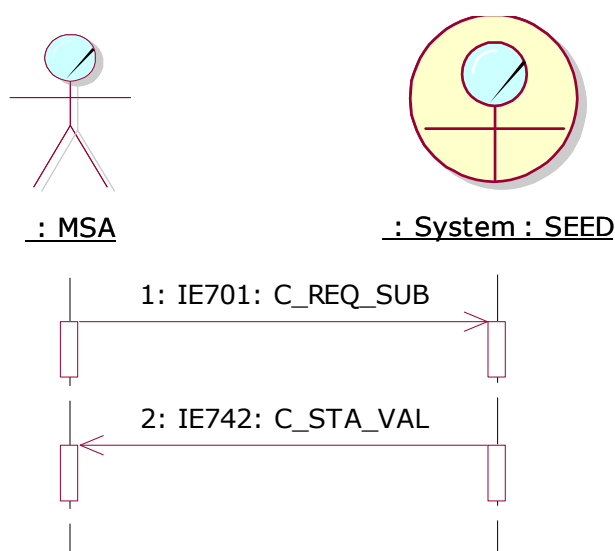


Figure 218: SEED statistics

Sub-Section V.II CS/MISE

The Central Services/Management Information System for Excise (CS/MISE) is located in the Common Domain. It provides the facilities to assist the monitoring and the reporting on the operations of EMCS. This is performed by collecting, distributing and publishing EMCS business and technical statistics, by providing information on movements (ARC follow-up⁶), and by monitoring the availability of all MSA applications (Availability Monitoring).

⁶ The ARC follow-up functionality provided by CS/MISE is based on the ARC nursing requirements described in Section X Transport of Messages via CCN/CSI.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.II.1 Messages involved

The information that is going to be exchanged regarding CS/MISE concerns only Availability Monitoring.

Three possible types of business process unavailability are defined:

- **Scheduled:** An unavailability planned in advance by an NEA for any implemented and open business service having specific begin and end date and time;
- **Unscheduled:** An unplanned unavailability by an NEA for any implemented and open business service having specific begin date and time and optionally end date and time;
- **Non-Implemented:** An unavailability that reports the non-implementation of a particular business service.

The Information Exchanges planned are:

- The preparation and dispatch by MSAs of the Business Process Unavailability Schedule C_AVA_DAT (IE770) to CS/MISE as identified in the process thread UC-007-110 [A1].
- The reception from CS/MISE of the Business Process Unavailability Schedule C_AVA_DAT (IE770) sent by the MSAs as identified in the process thread UC-007-310 [A1].
- The dissemination of the Business Process Unavailability Schedule C_AVA_DAT (IE770) from CS/MISE to the MSAs as identified in the process thread UC-007-310 [A1].
- The XML NACK C_XML_NCK (IE917) is used for reporting syntactic errors of the Business Process Unavailability Schedule C_AVA_DAT (IE770).

V.II.2 Manage unavailability (UC0.07)

Use case UC0.07 of FESS [A1] defines the message-based registration of unavailability decomposed into two scenarios:

- One scenario where a NEA registers to CS/MISE unavailability via sending an IE770 message (see V.II.2.1).
- One scenario where CS/MISE disseminates unavailability to all NEA via sending an IE770 message (see V.II.2.2).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

When the NEA registers to CS/MISE unavailability, two exceptional scenarios are foreseen; in the first scenario CS/MISE receives a syntactically invalid IE770 (see V.II.2.3.1) and in the second scenario CS/MISE receives a semantically invalid IE770 (see V.II.2.3.2). Exceptional scenarios are only foreseen when CS/MISE receives an invalid IE770, the opposite case where an NEA receives an invalid IE770 from CS/MISE is not foreseen since CS/MISE will always send valid IE770 messages. The semantic validations that CS/MISE performs on all IE770 messages received are also described in this Section (see V.II.2.4).

V.II.2.1 Message-Based Registration of Unavailability

When a MSA wants to inform other MSAs about the unavailability of a business process using message-based exchanges, it prepares an unavailability schedule message (IE770:C_AVA_DAT). *In an unavailability schedule message (IE770:C_AVA_DAT), an unavailability can be registered for the following business processes (via the means of selecting the corresponding Business Process Code):*

- *All Functionality (i.e. all FESS business process);*
- *EMCS - Central Circuit – External Domain Functionality (i.e., all business processes of UC2.01, UC2.05, UC2.06, UC2.07, UC2.10, UC2.12, UC2.33, UC2.34, UC2.36, UC2.43, UC2.44 and UC2.46 of Section II of FESS);*
- *EMCS - Central Circuit – Common Domain Functionality (i.e. all business processes of Section II of FESS);*
- *EMCS - Follow-up and Collaboration Functionality (i.e. all business process of Section IV of FESS);*
- *National SEED Functionality (i.e. all business processes of Section III of FESS that are executed by the MSA central services, i.e. national SEED).*

Note: Indicative examples of technical issues that may be encountered and the correspondent Business Process Code that can be used by the MSAs in the IE770:C_AVA_DAT message in order to better reflect the unavailable business processes, will be included in the CS/MISE User Manual (UMN).

Upon completion of the message preparation, the MSA application is then sending the IE770 message to CS/MISE through CCN/CSI or by uploading it to CS/MISE via the User Interface.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

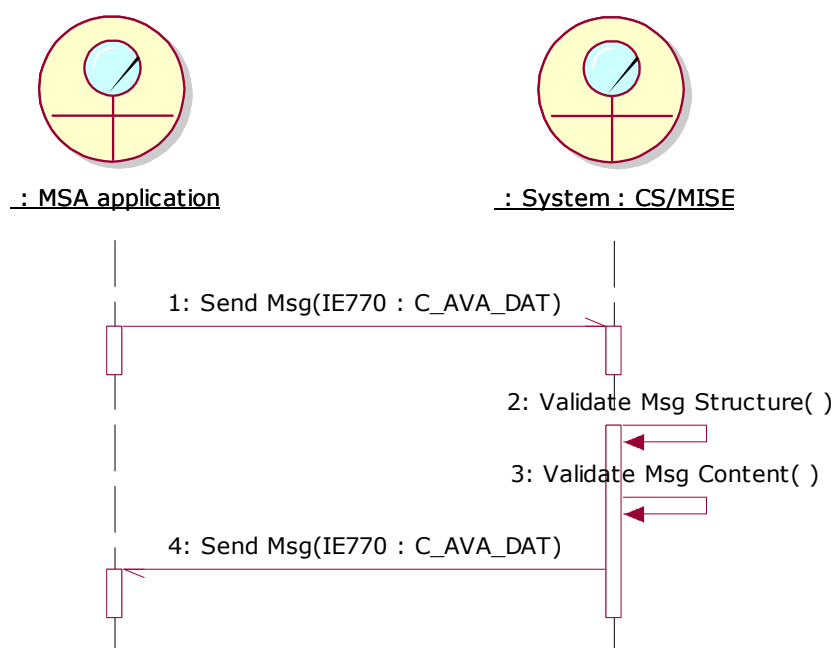


Figure 219: Sequence diagram for the announcement of unavailability

The CS/MISE validates the message structure and content, which in this scenario it is assumed to be found valid (Exceptional Scenarios are described Section V.II.2.3). Following, CS/MISE registers the unavailability information of the IE770 message into the system and sends the IE770 message back to the originating NEA for acknowledgement. This latter IE770 message sent by CS/MISE is correlated to the original IE770 message sent by the MSA application. Therefore, the <Correlation Identifier> Data Item of the latter IE770 message will have the value of the <Message identifier> Data Item of the original IE770 message. Moreover, due to the minimum resolution of time values in CS/MISE being minutes, if the original IE770 message contains seconds or even milliseconds, the IE770 message sent by CS/MISE will not contain them (milliseconds are removed and seconds are always zero).

The Sequence diagram for the announcement of unavailability to CS/MISE is presented in Figure 219 above. After the successful completion of this scenario, CS/MISE proceeds with the dissemination of the IE770 message (see V.II.2.2) to the MSA that opted for receiving IE770 disseminations except for the MSA that originally sent the IE770.

It should be noted, that if an IE770 message is uploaded via the CS/MISE User Interface, then the system will display a notification to the user confirming the successful registration of the IE770 message. Nevertheless, CS/MISE will still send the IE770 message back to the originating NEA as part of the dissemination process (see V.II.2.2), but –contrary to the case of registering unavailability through CCN/CSI– the IE770 message sent by CS/MISE will not be correlated to the original IE770 message uploaded via the CS/MISE User Interface.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.II.2.2 Dissemination of Unavailability

When a valid IE770 message is sent to CS/MISE through CCN/CSI or when an unavailability is registered/uploaded through the CS/MISE user interface, the system disseminates an IE770 message to all MSA that opted for receiving IE770 disseminations. The MSA that registered the concerned unavailability through CCN/CSI is excluded from the list of MSA that the IE770 message will be disseminated to since the MSA registering unavailability has already received the message as a confirmation (see V.II.2.1).

The reception of notifications for new unavailability of other Member States is configurable. Specifically, CS/MISE provides the option to define the mode of notifications reception for each Member State; the options are:

- through e-mail usually to the MSA Administrator;
- through the IE770 dissemination; or
- through both e-mail and IE770 dissemination.

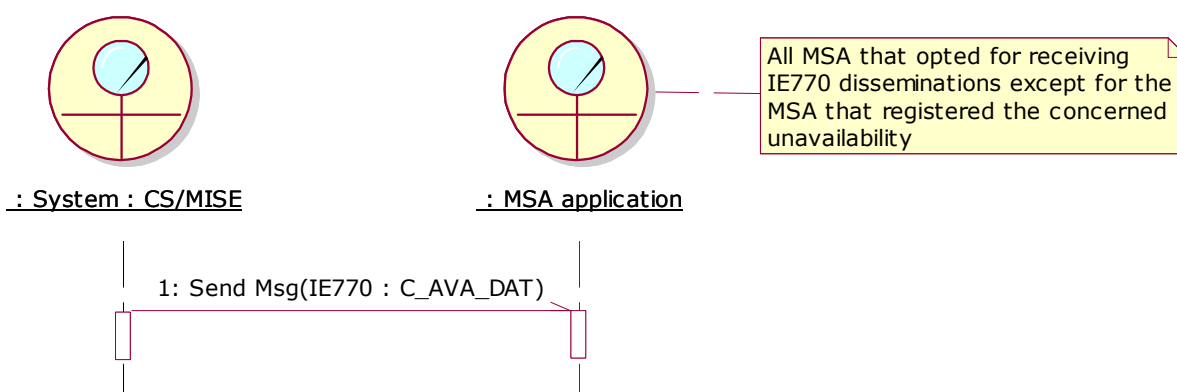


Figure 220: Sequence diagram for the dissemination of unavailability

For each newly registered unavailability, depending on the option used, a MSA may receive an e-mail, an IE770 message, or both.

The Sequence diagram for the dissemination of unavailability to all NEAs is presented in Figure 220 above.

V.II.2.3 Exceptional Scenarios

Two exceptional scenarios are identified based on the normal scenarios presented in the Sections V.II.2.1 and V.II.2.2 the identified scenarios are the following:

- a scenario where CS/MISE receives a syntactically invalid IE770 (see V.II.2.3.1);
- a scenario where CS/MISE receives a semantically invalid IE770 (see V.II.2.3.2).

It can be noticed that the exceptional scenarios concern only the cases where CS/MISE receives an invalid IE770. The opposite case where an NEA receives an invalid IE770 from CS/MISE is not foreseen since CS/MISE will always send valid IE770 messages.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

V.II.2.3.1 Syntactically Invalid Unavailability

CS/MISE always validates the message structure of all received IE770 messages for XML conformance and XSD compliance.

If any syntactic validation fails, then CS/MISE sends an IE917 (C_XML_NCK) to the NEA that originally sent the syntactically invalid message. In addition, syntactically invalid messages are not be processed by CS/MISE and no part of such messages is registered in CS/MISE, hence no changes to the Availability Schedule occur by syntactically invalid messages. This Sequence diagram for the announcement of a syntactically invalid unavailability to CS/MISE is presented in Figure 221 below.

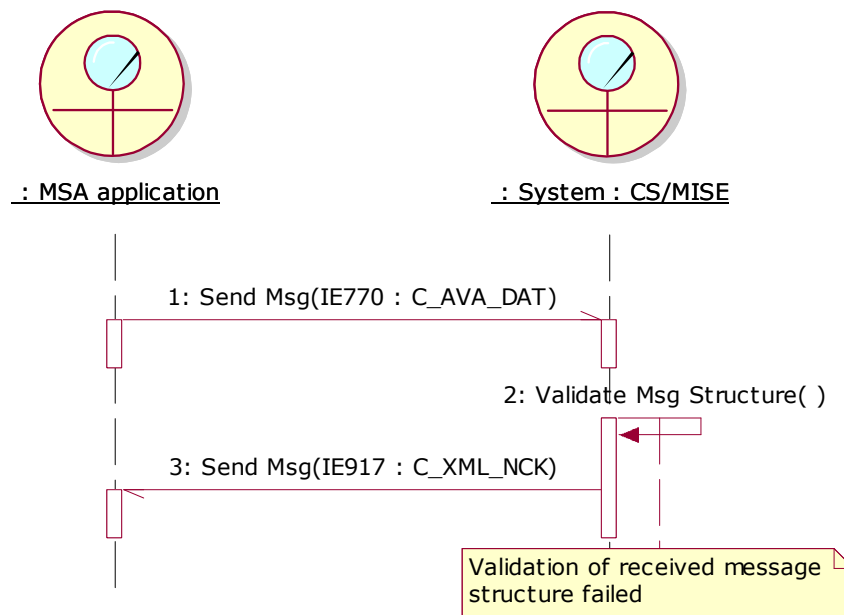


Figure 221: Sequence diagram for the announcement of a syntactically invalid unavailability

V.II.2.3.2 Semantically Invalid Unavailability

CS/MISE also performs semantic validation of received IE770 messages. If any semantic validation (see V.II.2.4) fails, CS/MISE sends an email to the Administrator of the NEA that originally sent the semantically invalid message. The semantically invalid messages are not processed by CS/MISE and no part of such messages will be registered in CS/MISE, hence no changes to the Availability Schedule occur by semantically invalid messages (i.e. registration of messages is atomic in the sense that it either succeeds or fails). The sequence diagram for the announcement of a semantically invalid unavailability to CS/MISE is presented in Figure 222 below.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

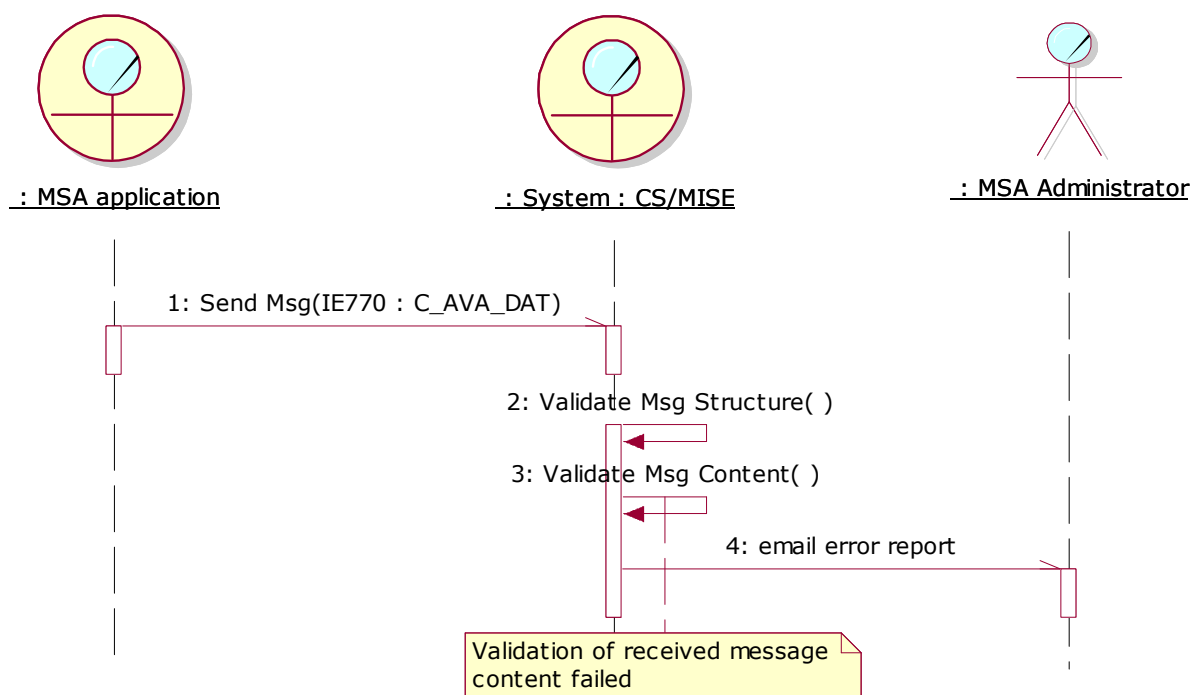


Figure 222: Sequence diagram for the announcement of a semantically invalid unavailability

V.II.2.4 Unavailability Semantic Validations

CS/MISE semantically validates incoming IE770 messages either received through the CCN/CSI or uploaded through the User Interface. In the case where a semantically invalid IE770 message is received through the CCN/CSI, the MSA Administrator is notified via email (see V.II.2.3.2). In the case where a semantically invalid IE770 message is uploaded through the User Interface, then the errors are displayed to the user on the User Interface.

CS/MISE raises a semantic validation error:

1. If the <Message Sender> Data Item does not contain a value of the form “NDEA.<Member State Code>”, where “<Member State Code>” is a value from the Member State Codes Business Codelist (this validation is in accordance to see Section VII.I.4 “Common message header”) that also corresponds to the MSA sending (or uploading) the IE770 message.
2. If the <Message Receiver> Data Item does not contain the value “CSMISE.EC” (when CS/MISE sends an IE770, it is the <Message Sender> that will have the value “CSMISE.EC”).
3. If another IE770 message from the same MSA and with the same <Message Identifier> value has already been received (this validation is in accordance to Section VIII.I.3.2.3.1 “Coordination protocol validations”).
4. If the value of the <HEADER>.<Unavailability Schedule Message Type> Data Item is “2: Notification to national domains” (the IE770 messages received by CS/MISE should have the value “1: Notification to common domain Central Services” and only the messages sent by CS/MISE should have the value “2: Notification to national domains”).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

5. If the value of the <UNAVAILABILITY SCHEDULE>.<Member State Code> Data Item is not a valid code of the Member States Codelist (BC11) or does not correspond to the MSA sending (or uploading) the IE770 message (see also semantic validation 1).

6. If the value of the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<Operation> Data Item is not a valid code according to rule R086, stating the following:

The <Operation> attribute is used to indicate what action is applied: ‘C’ (create), ‘D’ (delete).

In case of “updates”, the Data Group ‘SYSTEM UNAVAILABILITY’ to be updated will be first deleted and then (re-)created.

7. If the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<Operation> Data Item “C: Create”, but the contents of a <SYSTEM UNAVAILABILITY> descendant have the same business process code and the same starting date and time:

- with another <SYSTEM UNAVAILABILITY> within the same IE770 message (i.e. not necessarily a descendant of the same <(US) ACTION> or <UNAVAILABILITY SCHEDULE>), or
- with another entry of the Availability Schedule of CS/MISE (i.e. an entry has already been registered in CS/MISE via any means) concerning the same Member State and being the Scheduled or the Unscheduled type.

8. If the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<Operation> Data Item “C: Create”, but the contents of a <SYSTEM UNAVAILABILITY> descendant have the same business process code overlap:

- with another <SYSTEM UNAVAILABILITY> of Non-implemented type within the same IE770 message (i.e. not necessarily a descendant of the same <(US) ACTION> or <UNAVAILABILITY SCHEDULE>), or
- with another entry of the Availability Schedule of CS/MISE (i.e. an entry has already been registered in CS/MISE via any means) concerning the same Member State and being the Non-implemented type.

It is said that two unavailability entries overlap when their periods defined by the starting and ending date and time are overlapping.

9. If the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<Operation> Data Item “C: Create”, but the contents of a <SYSTEM UNAVAILABILITY> descendant overlap:

- with another <SYSTEM UNAVAILABILITY> of Non-implemented type having business process code “All” within the same IE770 message (i.e. not necessarily a descendant of the same <(US) ACTION> or <UNAVAILABILITY SCHEDULE>), or

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section V - Central Services	

- with another entry of the Availability Schedule of CS/MISE (i.e. an entry has already been registered in CS/MISE via any means) concerning the same Member State, being the Non-implemented type, and having business process code “All”.
10. If the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<Operation> Data Item “D: Delete”, but the contents of a <SYSTEM UNAVAILABILITY> descendant are not identical with an already existing entry in the Availability Schedule of CS/MISE (i.e. the same entry has not been registered in CS/MISE via any means).
 11. If the Data Item <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Type> is “S:Scheduled” and the Data Item <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Downtime End Date and Time> is missing, thus violating Condition C059, which states the following:

IF <Type> is "Scheduled"
THEN <Downtime TO> is ‘R’
ELSE <Downtime TO> is ‘O’
 12. If the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Downtime End Date and Time> exists and is earlier or equal to <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Downtime Start Date and Time>.
 13. If the value of the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Explanation> Data Item is provided and the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Explanation_LNG> Data Item is not a valid code of the Language Codes Codelist (BC12).
 14. If the Data Item <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Type> is “S:Scheduled” and the value of the <UNAVAILABILITY SCHEDULE>.<(US) ACTION>.<SYSTEM UNAVAILABILITY>.<Downtime Start Date and Time> is less than the 48 hours from the date and time that the IE770 message is processed.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VI - System Administration	

Section VI SYSTEM ADMINISTRATION

The procedures and tools (i.e. archiving procedures, configuration management, version control, data management, fallback procedures, problem tracking and audit trail) used for system administration are a national matter and consequently do not concern other MSAs.

NDEAs need to keep a log of exchange information, for relating an error to the information that has been exchanged and to solve any disputes regarding exchanged information. This log needs to contain:

- The content of the messages that have been exchanged (either sent or received), thus including all steering information specified by the Data Group MESSAGE (see Section VIII: Design Principles), e.g. ARC, Correlation identification, message reference and the functional message type;
- A timestamp showing at which date and time the Information Exchange (IE) has been prepared for sending or has been received;
- The result of application processing of the message, including all detected errors and any state change triggered by the message. If any error has been detected, the message will be viewed as not being processed by a national excise administrator;
- The name of the queue to which the message has been submitted;
- A timestamp showing the date and time at which the Information Exchange (IE) message has been exchanged through CCN. At reception, the timestamp is the date and time of receiving the Information Exchange (IE). At sending it is date and time of delivering the message to the CCN Gateway of the sending National Excise Application;
- The CSI header for the particular message. The structure of the CSI header is for instance given in the CCN/CSI Common Definition Reference Manual (C language, jCSI);
- The status of the exchange of the Information Exchange (IE) interchange across CCN (i.e. the reception of a Confirm on Delivery and its related timestamp).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VII - Technical Message Structure	

Section VII TECHNICAL MESSAGE STRUCTURE

VII.I.1 Introduction

In this section the basic elements of the EMCS Technical Message Structure, namely Data Groups, Data Items and Codelists, are described.

VII.I.2 Data dictionary

VII.I.2.1 Data Items

The different Data Items that are part of EMCS are listed in Appendix G. Every Data Item is identified by a unique name. The naming conventions are listed in Section I of this document.

Every Data Item has an associated type (as defined in section VIII.I.2.1.1 Data Item conventions) and in some cases a Data Item can have only discrete values, in which case the Data Item is said to have an associated Codelist.

In some cases, there may be some deviations between names at the corresponding FESS [A1], and DDNEA level. The reasons for this are discussed later in this document.

VII.I.2.2 Data Groups

The different Data Groups being part of EMCS are listed in Appendix F.

Every Data Group consists of a number of Data Items in a particular order. Every message is composed of a certain number of Data Groups in a particular hierarchy. Every Data Group is identified by a name. To be noted is that group names are not unique. It may thus very well happen that the same group name is found in different messages. Moreover, Data Groups with the same name do not always include the same Data Items. Hence, when a Data Group is used in more than one place including different Data Items each time, then this Data Group should be assigned to all of these Data Items even if not all of the Data Items are used in every instance.

Later on, with the message definition, the Data Items which are to be included for a particular group or not are specified.

Every Data Item within a group also has a unique identifier. The unique identifiers have been chosen more or less arbitrarily.

To be noted is that some Data Groups may not always have the same Data Item sequence in different messages.

Later on, with the message definition, the exact sequence of data elements in a particular group is specified.

VII.I.2.3 Codelists

A Codelist is a set of discrete values, with an associated meaning. Codelists are included in Appendix B.

A name and a number identify Codelists.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VII - Technical Message Structure	

There is number of Codelists for which the values are business dependent, hence mutable. These Codelists are known as business Codelists and are maintained by the SEED. The Central Project Team, supported by the Legal and Procedural team, will maintain the business Codelists on the central reference site. The MSAs can then download the new Codelists from this reference site. The values of the business Codelists can be found in the FESS [A1] Appendix B.

In addition, there are a number of technical Codelists for which the values are predefined and fixed. These values are not maintained within the common Reference Data. These Codelists are marked as technical Codelists in Appendix B.

VII.I.3 Technical message structure

The structure and format of the different Information Exchanges are included in the corresponding Appendix D. These appendices contain a message format description for every Information Exchange that is part of the particular system.

The technical message description is supplied in two parts.

The first part is the overall message description. This description contains the overall layout of the messages. It defines the different Data Groups that are part of the message, the sequence of the groups, the level of hierarchy of the Data Groups, the optionality of the Data Group, the possible repeat count, and associated rules and conditions. Concerning the optionality, it should be noted that the following rules apply:

- If a Data Group is always required, it is marked as “R”;
- If there exist one or more conditions related to the presence of the Data Group, it is marked as ‘D’. When a condition indicates that a dependent Data Group “does not apply” in a specified case then the specific Data Group must not be present in the message structure;
- If a Data Group is not always required and there are no conditions related to its presence, it is marked as ‘O’, meaning that the Data Group may either be present in the message structure or not. However, if information is available it is recommended to be included in the message despite the fact that this Data Group is characterised as Optional.

In order to go down one level in the hierarchy, the Data Group at the higher level in the hierarchy needs to be present. The second part of the TMS contains the description of the different Data Items. This description includes the sequence of the data elements in the group, the optionality, and the associated rules and conditions.

Concerning the optionality of the Data Items, the following rules apply:

- If a Data Item is always required, it is marked as “R”;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VII - Technical Message Structure	

- If there exist one or more conditions related to the presence of the Data Item, it is marked as ‘D’. When a condition indicates that a dependent Data Item “does not apply” in a specified case, then the specific Data Item must not be present in the message structure. It shall be noted that a Data Item set to “NULL” (empty valued) is still present in the message structure. Hence, when a Data Item is marked as “does not apply” in a specific case, the generated messages shall not include it with a “NULL” value. Subsequently, when all Data Items of a Data Group are set to “NULL”, the Data Group is still present in the message structure. Hence, when a Data Group is marked as “does not apply” in a specific case, it shall not be present in the generated messages with all of its Data Items set to “NULL”.
- If there are no conditions related to the presence of a particular Data Item, it is marked as ‘O’, meaning that the Data Group may either be present in the message structure or not. However, if information is available it is recommended to be included in the message despite the fact that this Data Group is characterised as optional.

The rules and conditions of FESS, Appendix D [A1] have been copied and marked as RXXX and CYYY. However, those Rules that include constant values or refer to the business entities (FESS, Appendix B: List of codes) have been replaced by technical and business Codelists in DDNEA, Appendix A. Moreover, when data is derived from another message, the rules and conditions are implicitly carried forward.

The message description part of this document consists of message hierarchies and correlation tables in order to map the Information Exchanges to those hierarchies (XML).

The mapping between the optionality codes used for the Data Groups and Data Items in the FMS of FESS [A1] and the ones used in the DDNEA is shown below:

Status description	FESS status code	DDNEA status code
Required/mandatory	R	R
Optional	O	O
Conditional /Dependent	C	D

Table 30: Use of status codes

The abbreviations stand for Required, Optional, Conditional/Dependent. The DDNEA optionality codes are used in the Correlation tables of Appendix C and in the TMS of Appendix D.

It should be noted that the Data Items are characterised as “Dependent” in the DDNEA instead of “Conditional”, as FESS does, since it is more accurate representation of the relationship between the elements and their values. This is due to the fact that not only *Conditions* but also *Rules* and *Technical Rules* can also be applied to these Data Items. Therefore, the letter “D” illustrates precisely the dependency on different entities. An indicative example extracted from Appendix D of the DDNEA is the following:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VII - Technical Message Structure	

The Data Item (*DELIVERY PLACE*) *TRADER*. *Trader ID* is dependent (D) since both the C074 and R045 are applied to it. According to C074, the presentation of this Data Item (if it is “Required” or not) is dependent on the *DESTINATION TYPE CODE* value. Moreover, according to R045, the content of this field (if it will be the excise number, the temporary authorisation, VAT number, etc.) is dependent on the *DESTINATION TYPE CODE*.

VII.I.4 Common message header

Each message is an envelope encapsulating a “Common Message Header” and a “Body” component.

The “Common Message Header” is commonly defined for all the messages and consists of the following Data Items:

- Message sender and Message recipient: The Data Items "Message sender" and "Message recipient" are both “Required” and should contain, respectively, the address of the sender and recipient, defined as:

<Application Name>.<Member State Code>, where:

- <Application Name> equals:
 - for NDEAs, the value “NDEA”;
 - for SEED, the value “SEED”;
 - for CS/MISE, the value “CSMISE”;
- <Member State Code> equals:
 - for NDEAs, the respective “Member State Code” value from the Member State Codes Business Codelist;
 - for SEED and CS/MISE, the value “EC”;
- Date of preparation: This is a “Required” Data Item used for the date that the Information Exchange was put into XML representation (generation of the XML message);
- Time of preparation: This is a “Required” Data Item used for the exact time that the Information Exchange was put into XML representation (generation of the XML message);
- Message identifier: This is a “Required” Data Item generated by the sending application to uniquely identify the information exchange;
- Correlation identifier: This is a “Dependent” Data Item, since it is only Required for correlating the response and refusal messages. It does not apply for requests and one way messages.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VII - Technical Message Structure	

The “Body” contains the actual business message. It is separately defined for each of the messages and consists of the rest of the elements comprising the business content of the messages.

VII.I.5 DDNEA consistency

The Information Exchanges are aligned with the FESS (Appendix D).

As a general overview, the major changes between DDNEA TMS and FESS FMS are:

- Changes in the naming conventions, some names have been changed between DDNEA and FESS [A1];
- Expansion of Information Exchanges inside other Information Exchanges: FESS [A1] presents some messages inside messages. In DDNEA, the content of the sub-messages has been put in the master-message;
- Message group: This Data Group is added in every message;
- Implementation of FMS conditions and rules;
- Technical Rules and Conditions.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

Section VIII DESIGN PRINCIPLES

This section defines a number of design principles for the NDEAs that are common regardless the transportation mechanism (CCN/CSI or SOAP/HTTP).

VIII.I.1 Approach

Every Information Exchange needs to be in a structure that conforms the TMS structure in Section VII - Technical Message Structure. The TMS needs to be formatted in XML format, as specified within this document in Section IX - XML formatting.

The formatted message needs to be transported across CCN/CSI and across SOAP/HTTP according to the rules laid out in this document (Section X - Transport of Messages via CCN/CSI and Section XI - Transport of Messages via SOAP/HTTP).

This applies only to mandatory exchanges. For the (strongly) recommended exchanges, it is highly advised to use similar conventions and rules.

Due to the fact that Information Exchanges are used to update data of Excise operations held by different applications, not all data is uniquely identifiable, and therefore, the following rules are applied for the updating of operation data:

- Key fields: The ARC is a key to uniquely identify EMCS operations. It is unique and it refers to EMCS movements. Each e-AD body is uniquely identified by its “Body Record Unique Reference” number within an ARC;

VIII.I.2 Character Sets and Data Item Conventions

Every MSA may maintain locally different character set(s) and Data Item conventions. These have usually been selected in order to best fit the MSA’s business needs. Excise does not impose any standards concerning national usage of character sets or Data Item conventions. However, they impose some standards for the Data Items (see VIII.I.2.1.1 Data Item conventions) and the character sets (see VIII.I.2.1.2 Character set usage) when information is exchanged in the Common Domain.

Every MSA should therefore foresee character set conversion and Data Item conversion when information is exchanged across the Common Domain.

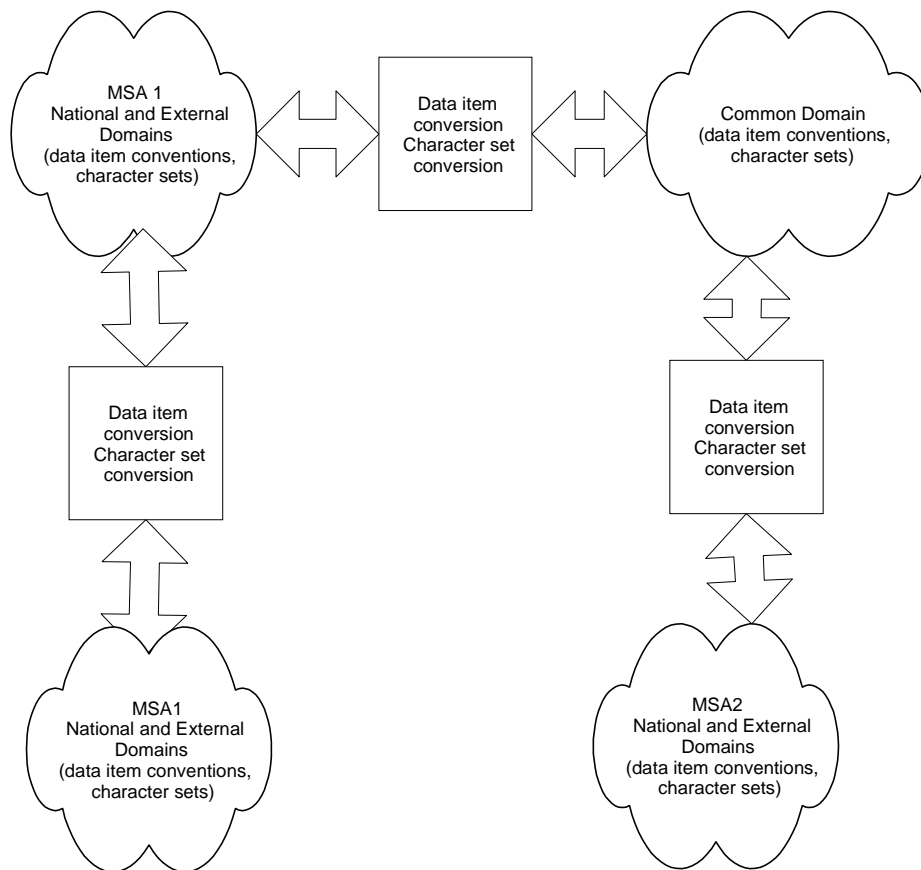


Figure 223: Character sets and conventions in use

The Common Domain standards are given below. Recommendations for National Domain exchanges are given next.

VIII.I.2.1 Common Domain exchanges

VIII.I.2.1.1 Data Item conventions

Every Data Item within a Common Domain Exchange can be numeric, alphanumeric, text field, dateTime, date, time or binary data. This Section defines the conventions used for the Data Item formats when present in the Common Domain. The Data Item conventions of the present Section are not part of the Syntactic layer validations, except for those incorporated in EMCS XSDs (Appendix H and Appendix I). It is the responsibility of the application sending message exchanges over the Common Domain (NEA and CDEA) to conform to the Data Item conventions defined in this Section.

VIII.I.2.1.1.1 Numerical Fields

Concerning numerical fields, it should be noted that these contain either a cardinal value (positive integer value) or a decimal value.

The decimal separator is the decimal point “.”. No other symbols are permitted as decimal separator.

Triad separators, such as a comma, shall not be used.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

Signs, whether positive or negative, shall not be used (all values are always intrinsically positive).

For decimal values, the decimal notation (with the decimal point) should only be used when there is a reason to indicate precision.

E.g., for a mass value:

- 89 kg, with a precision of 1 kg.
- 89.2 kg, with a precision of 0.1 kg.
- 89.20 kg, with a precision of 0.01 kg.

For numerical values, leading zeroes shall not be used. Trailing zeroes should only be used to indicate precision.

If the decimal point is present, at least one digit shall be present before the decimal point.

If the decimal point is present, at least one digit shall be present after the decimal point.

Examples for a n..11,3 type.

12345678.123	(Valid).
123456789012.123	(Invalid - too many digits before decimal point and hence too many digits in total).
12345678.1234	(Invalid - too many digits after decimal point and hence too many digits in total).
0123	(Invalid - leading zero not permitted).
+123	(Invalid - plus sign not allowed).
-123	(Invalid - minus sign not allowed).
1,234	(Invalid - triad separator not allowed).
.3	(Invalid - no digit before decimal point).
12345.	(Invalid - no digit after decimal point).
0.3	(Valid).
1.3E1	(Invalid - only digits and decimal point allowed).
12345678901	(Valid - n..11,3 can have maximally 11 digits of which maximally 3 after decimal points).

It is to be noted that the rules above also apply to numerical values within Codelists. Values in such a list should always be stored without leading zeroes (in order to avoid problems of comparing e.g. a value of 60 against a value of 060). If the leading zeroes are omitted, a numerical comparison should always work out fine.

It should be noticed that there are no Codelists with decimal values.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

VIII.I.2.1.1.2 Date/Time Fields

The specification of Date and/or Time fields used in TMS (Section VII and Appendix D) is as per W3C XML Schema specification [S15] except that:

- for all times in DateTime and Time fields time zone must be omitted with the local time always implied as being the Coordinated Universal Time (UTC, sometimes called "Greenwich Mean Time");
- all years in DateTime and Date fields are in the Common Era (i.e. AD), hence the negative sign is not permitted.

Although the reader should refer to the W3C XML Schema specification [S15], the following table (Table 31) indicates the format for each type and their corresponding regular expression.

Type	Regular Expression
Date	yyyy '-' MM '-' dd \d{4}-\d{2}-\d{2}
Time	hh ':' mm ':' ss ('.' s+)? \d{2}:\d{2}:\d{2}(\.\d+)?
Date/Time	yyyy '-' MM '-' dd 'T' hh ':' mm ':' ss ('.' s+)? \d{4}-\d{2}-\d{2}T\d{2}:\d{2}:\d{2}(\.\d+)?

Table 31: Date/Time fields format and their corresponding regular expressions

Where:

- yyyy is a four-or-more digit;
- the remaining '-'s are separators between parts of the date portion;
- MM is a two-digit numeral that represents the month;
- dd is a two-digit numeral that represents the day;
- 'T' is a separator indicating that time-of-day follows;
- hh is a two-digit numeral that represents the hour; '24' is permitted if the minutes and seconds represented are zero, and the dateTime value so represented is the first instant of the following day (the hour property of a dateTime object in the .value space cannot have a value greater than 23);
- ':' is a separator between parts of the time-of-day portion;
- mm is a two-digit numeral that represents the minute;
- ss is a two-integer-digit numeral that represents the whole seconds;
- '.' s+ (if present) represents the fractional seconds.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

VIII.I.2.1.1.3 Binary data

For Data Items containing arbitrary binary data, the Base64-encoding will be used, thus the base64Binary XML Schema datatype.

The specification of base64Binary fields used in TMS (Section VII and Appendix D) is as per W3C XML Schema specification [S15].

It shall be noted that the file format of binary embedded data should be limited to two specific standards based file formats, as defined in TESS [A11]:

- Joint Photographic Experts Group (JPEG) for images;
- Portable Document Format (PDF) for documents.

The file format of binary embedded data can be determined by the first few bytes of the file (also known as “magic number”):

- The Joint Photographic Experts Group (JPEG) files always begin with the bytes “FF D8”;
- The Portable Document Format (PDF) files always begin with the bytes “25 50 44 46” (the hex values of “%PDF”).

VIII.I.2.1.1.4 Language Indicator for Language-sensitive text fields

The Language Indicator (LNG), associated with Language-Sensitive Data Items, denotes the language that the text was intended to be written. This does not preclude cases where characters of another language (i.e. other than the one indicated) are present in the associated Language-Sensitive text field. It should be noted that the character set and the encoding used in EMCS (Unicode and UTF-8, respectively, as per VIII.I.2.1.2 Character set usage) supports characters of multiple languages.

VIII.I.2.1.2 Character set usage

The character set used in all Common Domain message exchanges is the Unicode Standard [S2]. The encoding used in all Common Domain message exchanges is the UTF-8 Standard [S3]. It should be noted that both of the aforementioned requirements concerns the whole message content.

VIII.I.2.2 External Domain exchanges

It is highly recommended to use the standards defined above, to the maximum extent possible, in the External Domain.

VIII.I.3 Exception Handling

VIII.I.3.1 Introduction

Exceptions may occur in any message exchange between EMCS applications. In order to support the business processes defined in FESS [A1], the DDNEA defines how EMCS applications should handle exceptions occurring at the following message exchanges:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

- NEA to NEA at the Common Domain;
- NEA to SEED at the Common Domain; and
- Economic Operator to NEA at the External Domain (exception handling at the External Domain should be considered as a recommendation subject to national adaptation).

The fallback and recovery procedures of these message exchanges are outside the scope of DDNEA. In principle, every action related to Information Exchanges needs to be logged to allow recovery and identification of a failed component. Relevant procedures are specified in FRS [A12].

It should be noted that refusal and rejection messages are two distinct categories of messages and should not be confused. All messages relevant to exception handling are refusal messages originating automatically from the EMCS applications denoting that processing of a received message has been refused. A message sender receiving a refusal message as a response to a message sent, should consider that message as non-processed by the receiver.

Rejection messages, on the other hand, are triggered by a business process and are described by the business scenarios. Rejection messages have corresponding state transitions and are not response messages. This section does not discuss rejection message since they are discussed in the business scenarios of the DDNEA.

VIII.I.3.2 Exception hierarchy

The validations discussed in this section follow a hierarchy constituted by a number of layers (see Figure 224); each successive layer is more abstract than the previous layer. Each layer's validations should pass successfully in order to proceed to the validations of the next layer. Depending on the layer at which an exception is raised, particular actions should be taken by the message receiver in order to inform the sender of the exception and the refusal to process the message.

The exception hierarchy consists of the following layers presented in the order that should be performed:

1. transport layer: for validations and errors during sending or receiving messages over the Common Domain (see VIII.I.3.2.1);
2. syntactic layer: for validations of the message content for XML and XSD conformance (see VIII.I.3.2.2);
3. semantic layer: for validations relating to the coordination protocol and business compliance (see VIII.I.3.2.3).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

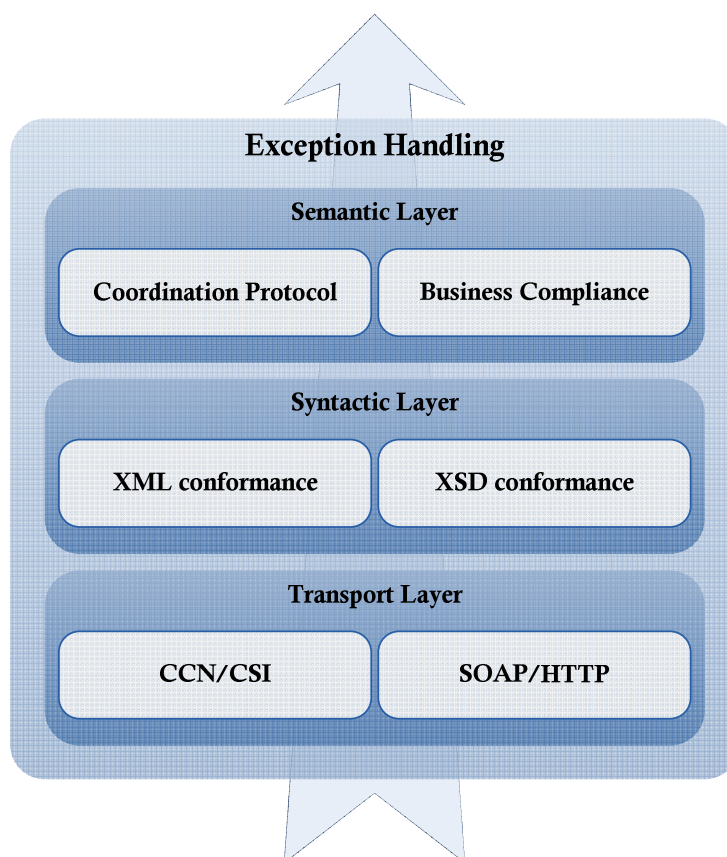


Figure 224: Exception Hierarchy

VIII.1.3.2.1 Transport layer

Exceptions occurring at the transport layer are basic exceptions that ultimately prohibit the receiver from retrieving the message sent by the sender. Depending on the transport being used, the exception handling is discussed in Section X for CCN/CSI and Section XI for SOAP/HTTP.

The validations of this layer are applicable to all Common Domain message exchanges. For the External Domain the validations of this layer are recommended, subject to the particular communication protocols used at national level.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

VIII.I.3.2.2 Syntactic layer

After reception of a Common Domain XML message and before the message is further processed the receiver validates the size of the received message, in order to avoid unresponsiveness or abrupt failure of the receiving application. More specifically, the size of the messages exchanged over the Common Domain should not exceed the threshold size limit which is set to 21 Mbytes (2^{20} bytes), as defined in TESS [A11]. In the case that the received message is within the acceptable size limit, the receiver validates that its content conforms to the XML standard 1.0 [S1] and that it is UTF-8 encoded as per VIII.I.2.1.2 Character set usage. Then, the receiver validates the structure of the received Common Domain XML message against the corresponding XSD defined in Appendix H (see also Section IX XML formatting). The aforementioned validations are defined as the Syntactic layer validations and when any of the aforementioned validations fails for a given Common Domain XML message, then that message is defined as syntactically invalid⁷.

The validations of this layer are applicable to all Common Domain message exchanges. For the External Domain the validations of this layer are highly recommended subject to the particular national requirements. Although, syntactic validations are recommended in the External Domain rather than required, the scenarios in the DDNEA consider this validation to be performed.

The syntactic validation of messages is explicitly depicted in the scenario diagrams of the DDNEA with the “Validate Msg Structure ()” operation. The purpose of this operation is to ensure that the message size, the XML formatting and the message structure conform to the DDNEA specifications. The “Validate Msg Structure ()” operation occurring in the scenario diagrams, is considered to return successfully meaning that the message is syntactically correct.

A message receiver of a syntactically invalid message will not be able to process the message. The normal processing of the invalid message should be aborted and the processing refusal should be indicated to the message sender by dispatching a Negative Acknowledgement of XML Receipt message (IE917: C_XML_NCK) containing the detected error(s).

The message sender after reception of the IE917 message should not retransmit the same message before correcting the errors. Moreover, the message sender should consider the referenced message as non-processed by the receiver, thus no changes at the business level should occur for both the message sender and the message receiver.

The IE917 message contains the following information:

- The character line and column number of the error identifying the first position of the identified location of the error within the message.

⁷ It should be noted that the conventions of Section VIII.I.2 Character Sets and Data Item Conventions, are not part of the Syntactic layer validations. However, (as defined in Section VIII.I.2 Character Sets and Data Item Conventions) it is the responsibility of the application sending message exchanges over the Common Domain (NEA and CDEA) to conform to the Data Item conventions defined in this Section.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

- The Error Reason containing the text of the error for which no specific requirements are imposed.
- Optionally, the Error Location containing the XPath location of the error. For message with size that exceeds the threshold size limit or with invalid XML content or if the reported error concerns XML conformance or formatting, the XPath location will, most probably, not be available in which case the Error Location needs not be included. Even for XML schema errors, this data item is optional as the sender's application may not support this information. Moreover, it is recommended to avoid including the Error Location if the XPath string is to be truncated (i.e. if the length of the string is greater than the length of the relative Data Item), thus avoiding reporting inaccurate location.
- Optionally, the Original Attribute Value that should be used when the error is an XML schema error concerning an invalid value. The reasons for considering an attribute value invalid might be the format or a value outside the applicable technical Codelist. For such cases, the Original Attribute Value should contain the invalid value contained in the received message in order to indicate which value was perceived invalid.

For cases where the received message contains Administrative Reference Code and Sequence Number and the information is syntactically valid (i.e. the XML content can be parsed), then the information should be included in the Negative Acknowledgement (IE917: C_XML_NCK) message to support correlation at the message sender's side. Therefore, the <HEADER> Data Group should be present. If the received message contains syntactically invalid Administrative Reference Code, the <HEADER> Data Group should not be present.

More specifically, the following cases are identified that complement Technical Rule TR0104 of Appendix D. When the received message is IE801, IE802, IE803, IE807, IE810, IE813, IE818, IE819, IE837, IE861, IE871, IE904, or IE905:

- If the received message contains a syntactically valid <Administrative Reference Code> Data Item:
- If the <Administrative Reference Code> Data Item is known (the message receiver has already received at least one message with the specific Administrative Reference Code):
 - If the received message contains a syntactically valid <Sequence Number> Data Item, then the <HEADER> Data Group should be present in the IE917 message (C_XML_NCK), the value of the <Administrative Reference Code> Data Item should equal to the received Administrative Reference Code, and the value of the <Sequence Number> Data Item should equal to the received Sequence Number;
 - If the received message contains a syntactically invalid or is missing (either erroneously missing or the received message is an IE810 or an IE807) <Sequence Number> Data Item, then the <HEADER> Data Group should be present in the IE917 message (C_XML_NCK), the value of the <Administrative Reference Code> Data Item should equal to the received Administrative Reference Code, and the value of the <Sequence Number> Data Item should equal to the value of the last known Sequence Number for the specific Administrative Reference Code;

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

- If the <Administrative Reference Code> Data Item is unknown (the message receiver hasn't received any message with the specific Administrative Reference Code):
 - If the received message contains a syntactically valid <Sequence Number> Data Item, then the <HEADER> Data Group should be present in the IE917 message (C_XML_NCK), the value of the <Administrative Reference Code> Data Item should equal to the received Administrative Reference Code, and the value of the <Sequence Number> Data Item should equal to the received Sequence Number;
 - If the received message contains a syntactically invalid or is missing (either erroneously missing or the received message is an IE810 or an IE807) <Sequence Number> Data Item, then the <HEADER> Data Group should be present in the IE917 message (C_XML_NCK), the value of the <Administrative Reference Code> Data Item should equal to the received Administrative Reference Code, and the value of the <Sequence Number> Data Item should equal to the value of "1";

If the received message contains a syntactically invalid <Administrative Reference Code> Data Item or is missing, then the <HEADER> Data Group should not be present in the IE917 message (C_XML_NCK).

VIII.1.3.2.3 Semantic layer

After successful syntactic validation, the format and structure of the message have been secured and thus the receiver may continue processing at the business level. Two categories of validations should be performed at the semantic layer, namely coordination protocol validations and business compliance validations. The coordination protocol validations ensure that the received message is expected as part of the message exchange protocols defined in the business scenarios of the DDNEA. Finally, the business validations ensure the business rules compliance of the received message such that the relevant business processes will be executed without the occurrence of any exceptions.

Depending on the business communication channel through which a message is received, not all categories of validations are applicable. The applicability of the semantic layer validation categories that the receiver should perform per business communication channel is depicted in Table 32.

	Economic Operator to NEA	NEA to NEA	NEA to SEED
Coordination protocol	Recommended if applicable	Mandatory	Mandatory
Business compliance	Recommended	Not applicable	Mandatory

Table 32: Semantic layer validations applicability to business communication channels

The recipients of NEA to NEA message exchanges should perform only coordination protocol validations since it is assumed that a NEA sending a message to another NEA has validated successfully the business compliance validity of the messages before transmission.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

It is required, therefore, for messages exchanged between NEAs to be valid. Towards achieving this requirement, it is highly recommended that NEAs perform business compliance validations when a message received from the Economic Operator to NEA business communication channel triggers a message transmission to another NEA through the Common Domain. The business compliance validation is explicitly depicted in the scenario diagrams of the DDNEA with the “Validate Msg Content ()” operation.

The “Validate Msg Content ()” operation is considered to return successfully, meaning that the message complies with all rules, conditions and functional requirements of the FESS [A1]. Again, this is subject to particular requirements at national level.

Finally, for NEA to SEED message exchanges, SEED will perform both categories of semantic validations.

Semantic errors are notified by different messages depending on the erroneous message. There are particular refusal messages for certain messages; for refusing an IE701, an IE702 is used (Sections III.I.13 General query to retrieve an e-AD (UC2.52) and V.I.2.1.2 Re-synchronisation of Reference Data (UC1.05)); for refusing an IE713, an IE714 is used (V.I.3.1.2 Dissemination of SEED data (UC1.14)). For all other cases, the generic Functional Negative Acknowledgement (IE906: C_FUN_NCK) message should be used for the Common Domain and the IE704 for the External Domain. The IE906 (C_FUN_NCK) message contains the following information:

- The Error Type data item containing the code of the error as a numerical value from the applicable Codelist.
- The Error Reason is an alphanumeric data item to include a human-readable description of the error. The data item might contain the description corresponding to the code used for the Error Type data item, or further reasons and/or explanations about the error. If the Error Type data item contains the value of zero, which does not have a specific description, the message issuer may further explain the reason for the error.
- Optionally, the Error Location containing the XPath location of the error. This data item is optional as the sender’s application may not support this information. It is recommended to avoid including the Error Location if the XPath string is to be truncated (i.e. if the length of the string is greater than the length of the relative Data Item), thus avoiding reporting inaccurate location.

It should be noted that since TR0104 applies on both the IE906 and IE917 message, the guidelines for the values that should be used for the <Sequence number> and the <Administrative Reference Code> Data Items of the <HEADER> Data Group, defined in Section VIII.I.3.2.2 Syntactic layer for the IE917 message, apply also to the respective Data Items of the IE906 message, with the exception of these guidelines applicable in the case of syntactically invalid <Administrative Reference Code> and/or <Sequence number>. Similarly, since C150 that applies on the <Sequence number> and the <Administrative Reference Code> Data Items included in the <HEADER> Data Group of the IE704 message follows the same logic with TR0104, the aforementioned guidelines applicable to the IE906 message should also apply for the values of these Data Items of the IE704 message.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

VIII.I.3.2.3.1 Coordination protocol validations

A number of validations relating to the coordination protocol should be performed in order to facilitate the exception handling in the Common Domain as described in the scenarios of Sub-Section III.VI Exception Handling.

The errors resulting from coordination protocol validations are summarised in Table 33 and the applicability of these errors per message are summarised in Table 34. A description for each error code follows the aforementioned tables.

Code	Description	Remarks
91	Duplicate LRN	The LRN of the received message is already known and is therefore not unique according to the specified rules.
26	Duplicate detected	The same interchange is received again. Duplication is detected by reception of an interchange reference that has already been received.
93	Invalid ARC	The structure of the ARC does not conform to specifications given in FESS [A1] Appendix B.
94	Invalid Follow Up Correlation ID / MV Correlation ID	The structure of the Follow Up Correlation ID or MV Correlation ID does not conform to specifications given in FESS [A1] Appendix B.
90	Unknown ARC	The ARC of the received message is not known, whereas it is expected to be known.
95	Unknown Follow Up Correlation ID / MV Correlation ID	The Follow Up Correlation ID or MV Correlation ID of the received message is not known, whereas it is expected to be known.
92	Message out of sequence	The message cannot be processed, because the receiver is not in a proper state.

Table 33: Coordination protocol error codes

	IE702	IE704	IE714	IE906
Duplicate LRN		✓		
Duplicate detected	✓	✓	✓	✓
Invalid ARC				✓
Invalid Follow Up Correlation ID / MV Correlation ID				✓
Unknown ARC				✓
Unknown Follow Up Correlation ID / MV Correlation ID				✓
Message out of sequence				✓

Table 34: Coordination protocol errors to refusal message map

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

Code 91 (Duplicate LRN) should be used for the External Domain and for IE704 only.

Code 26 (Duplicate detected) is of particular importance to the coordination protocol that ensures the uniqueness of the Message Identifier for a pair of sender/receiver as defined in VII.I.4 Common message header. For IE702 and IE714 only code 26 (Duplicate detected) from the coordination protocol error codes applies and it is only used by central SEED in the NEA to SEED message exchanges.

Code 93 (Invalid ARC) should be used for refusing received messages containing invalid values of the “Administrative Reference Code” Data Item. The fields that constitute the “Administrative Reference Code” Data Item should comply to the definition provided in Appendix B of FESS [A1].

Code 94 (Invalid Follow Up Correlation ID / MV Correlation ID) should be used for refusing received messages containing invalid values of the “Follow Up Correlation ID” or the “MV Correlation ID” Data Item. The fields that constitute the “Follow Up Correlation ID” or the “MV Correlation ID” Data Item should comply to the definition provided in Appendix B of FESS [A1].

It should be noted that the error codes 93 and 94 do not concern the format of the ARC, the Follow Up Correlation ID and the MV Correlation ID since this is covered by the syntactic layer validations through specialised regular expression patterns in the XSD files. The error codes 93 and 94 concern the validation of the second field (MS Identifier) of the ARC and the Follow Up Correlation ID or the MV Correlation ID respectively, in order to ensure that it is a valid code of the Member States codelist (defined in IE734: C_RDD_DAT message disseminated by SEED).

Codes 90, 95, and 92 concern the state-machines defined in previous sections of the DDNEA. The state-machines define states and particular messages that are allowed to be sent and expected to be received depending on the role that each application assumes and the state that is found to be in. An application receiving a message that ought to cause a state transition, but the message is unexpected according to the state-machines should report to the message sender a coordination protocol violation with an Functional Negative Acknowledgement (IE906: C_FUN_NCK) message.

Codes 90 (Unknown ARC) and 95 (Unknown Follow Up Correlation ID / MV Correlation ID) should be used for the cases where the application receives a message bearing an identifier (ARC or Follow Up Correlation ID or MV Correlation ID) received for the first time, but according to the respective state-machine the message is not included in the message(s) expected to be received from the initial state.

If an application receives a message bearing an identifier (ARC or Follow Up Correlation ID or MV Correlation ID) that is known, but the message is unexpected according to the state-machines (including the conditions for the Sequence Number when the identifier is the ARC) it should be refused with an Functional Negative Acknowledgement (IE906: C_FUN_NCK) message indicating a “Message out of sequence” error (Code 92).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

VIII.I.3.2.3.2 Business compliance validations

A number of validations relating to the business compliance should be performed in order to facilitate the exception handling in the Common Domain as described in the scenarios of Sub-Section III.VI Exception Handling. The errors resulting from business compliance validations are summarised in Table 35.

Code	Description	Remarks
0	Other	This value should be used only when the error does not correspond to any of the following values. It must be stressed that the value of zero should only be used for intra-release migration issues.
12	Incorrect (code) value	Value of an element in a message is outside the applicable business codelist.
15	Not supported in this position	The element or value is not allowed according to the applicable Rule(s) or Condition(s).

Table 35: Business compliance error codes

It should be noted that in the NEA to NEA message exchanges, the business compliance validations should not be performed. The IE702, IE704 and IE714, however, should perform all validations of Table 35. The applicability of business compliance errors per message are summarised in Table 36.

	IE702	IE704	IE714	IE906
Other	✓	✓	✓	✓
Incorrect (code) value	✓	✓	✓	
Not supported in this position	✓	✓	✓	

Table 36: Business compliance errors to refusal message map

The IE714 error codes contain specific codes for refusing economic operator updates presented in Table 37.

Code	Description	Remarks
0	Other	This value should be used only when the error does not correspond to any of the following values. It must be stressed that the value of zero should only be used for intra-release migration issues.
8	Trader Authorisation already exists (creation)	A record with the specified Trader Excise Number already exists.
9	Tax warehouse already exists (creation)	A record with the specified Reference of Tax Warehouse already exists.
10	Temporary authorisation already exists (creation)	A record with the specified Temporary Authorisation Reference already exists.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

Code	Description	Remarks
11	Trader Authorisation not found (Update / Deletion)	The Update / Deletion operation cannot be performed since the specified Trader Excise Number cannot be found.
12	Tax warehouse not found (Update / Deletion)	The Update / Deletion operation cannot be performed since the specified Reference of Tax Warehouse cannot be found.
13	Temporary authorisation not found (Update / Deletion)	The Update / Deletion operation cannot be performed since the specified Temporary Authorisation Reference cannot be found.
27	Inconsistency between Excise number and Excise office	The specified Excise office cannot be used for the specified Excise number (applies to all authorisations).
41	Only a warehouse keeper may be allowed to use a tax warehouse	The data group (USING) TAX WAREHOUSE only applies to tax warehouse keepers.
42	Invalid Reference of Tax Warehouse (R204 violation)	The specified Tax Warehouse Reference is not a valid <TAX WAREHOUSE.Reference of Tax Warehouse> or it is invalid according to Rule R204.
43	Missing Authorised Warehouse Keeper referencing Tax Warehouse (R205 violation)	The specified Tax Warehouse Reference is not a valid <(USING) TAX WAREHOUSE.Reference of Tax Warehouse> defined in Rule R205.
44	<Trader Excise Number> is missing (Cond157 violation)	The Consignor Trader Excise Number is not specified for the Temporary Authorisation that is not flagged as Small Wine Producer.
45	Invalid value for <Excise Product Code> (Rule212 violation)	The specified Excise Product Codes are other than 'W200' and 'W300' for the Temporary Authorisation that is flagged as Small Wine Producer.
46	The Member State of the Temporary Authorisation and the declared Consignor is the same (R233 violation)	The Member State of the Consignor is the same with the Member State the Temporary Authorisation is registered for.
112	Incorrect (code) value	Value of an element in a message is outside the applicable business codelist.
115	Not supported in this position	The element or value is not allowed according to the applicable Rule(s) or Condition(s).

Table 37: Business compliance error codes specific to IE714

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

The IE702 error codes contain specific codes for refusing common requests presented in Table 38. Again, it should be noted that in the NEA to NEA message exchanges, the business compliance validations should not be performed. The only IE702 business compliance error codes applicable to the NEA to NEA message exchanges are codes 2 and 8 used in the scenarios of Section III.I.13 General query to retrieve an e-AD (UC2.52).

Code	Description	Remarks
0	Other	This value should be used only when the error does not correspond to any of the following values. It must be stressed that the value of zero should only be used for intra-release migration issues.
2	No e-AD(s) retrieved matching selection criteria	
3	Reference data not available	
4	Excise Office List not available	
5	SEED data not available	
7	Unknown requested data	
8	Increment number out of range	Also used when the maximum limit of retrieved e-ADs has been reached.
112	Incorrect (code) value	Value of an element in a message is outside the applicable business codelist.
115	Not supported in this position	The element or value is not allowed according to the applicable Rule(s) or Condition(s).

Table 38: Business compliance error codes specific to IE702

VIII.I.4 Availability and Performance Constraints

The requirements for systems availability for NDEAs, only consider the international requirements, i.e. the NDEAs, availability required to support interoperability with other participating MSAs through the Common Domain. This also means that this constraint does not apply to the External Domains.

During Initial Implementation, all participating MSAs have to guarantee that in the event of a failure of their National Excise Application (NEA) or when their NEA is deliberately taken off-line because of business or technical requirements, all Information Exchanges received from CCN will be held until the NEA comes back on-line. They can then be processed as normal, subject to the NEA fallback rules [A12].

This minimum level of availability is met by the CCN functionality.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section VIII - Design Principles	

Sending NEAs should not re-send IEs for which there have been no replies as a result of unavailability of the receiving NEA. The re-sending should occur only upon request of the NEA, or receipt of an exception report.

Specifically for NEAs, the sending NEA should “enable” the sequence check mechanism in order to respect the correct sequence of messages, when this is explicitly defined, as is the case for IE801 and IE818.

Performance constraints related to Information Exchanges crossing the External Domain (communication between a NEA and its Traders) or within the National Domain (communications between different locations of the same NEA) are a national matter and should be fixed individually by each NEA.

The time constraints applied to Information Exchanges crossing the Common Domain (communication between two NEAs or between a NEA and Central EMCS Reference Site) must permit meeting the performance constraints defined in Appendix A of FESS [A1].

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IX - XML formatting	

Section IX XML FORMATTING

This section defines how the messages need to be formatted in XML format. In particular, it specifies the XML conventions for EMCS as well as the Character Sets that shall be supported by NDEAs. Finally, the XML mapping of IEs and the XSDs are defined in Appendix E and Appendix H respectively.

IX.I.1 XML Schema

The XML (Extensible Markup Language) is a subset of SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, its goal is to enable generic SGML to be served, received and processed on the web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML. The complete specific of XML 1.0 can be found at <http://www.w3.org/TR/1998/REC-xml-19980210>⁸.

The XML schema definition language [S15] is a model for describing the structure of information. In particular, the XSD defines a type of XML document in terms of constraints upon what elements and attributes may appear, their relationship to each other and what types of data may be in them. The schema language, which is introduced in XML 1.0 and uses namespaces, considerably extends the capabilities found in XML 1.0 document type definitions (DTDs).

The Design Document for National Excise Applications imposes that XML formatted messages are valid, UTF-8 encoded [S3], XML 1.0 [S1] documents that follow the rules defined in the XML Schema [S15].

IX.I.1.1 XML namespaces

The XML format specification makes use of the namespace concept. XML namespaces provide a simple method for qualifying element and attribute names used in XML documents. The use of different namespaces visually identifies the source schema for a XML element in an instance and enables the logical separation of elements that represent the XML Entity Model, XML Entity Actions and XML message definition.

The namespace used to qualify the repository of the XML Schemas of the Information Exchanges is the “IE”. The XML Schemas of the Information Exchanges reference other XML Schemas, henceforth refer to as Helper Schemas, with the following namespaces:

- “EMCS” to qualify the EMCS type definitions related to the XML Information Exchanges;
- “TMS” to qualify the EMCS Common Message Header elements and attributes related to the XML Information Exchanges;
- “TCL” to qualify the definition of the codes of the EMCS Technical Codelists related to the XML Information Exchanges;

⁸ An interesting annotated version of the specification is available at <http://www.xml.com/axml/testaxml.htm>.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IX - XML formatting	

- “DOC” to qualify the EMCS documentation elements related to the XML Information Exchanges.

Whenever DDNEA evolves, the Schemas of the messages might be affected. To ensure that the same Schema evolutions are used consistently by all EMCS applications, the Schemas are phased and versioned. The namespace of each Schema incorporates the applicable phase and version. All XML messages exchanged in EMCS include the namespace which they instantiate, thus all XML messages exchanged in EMCS refer to a particular phase and version.

An application will not be able to process a receiving XML message containing unknown namespace(s). In this case, the receiving application should dispatch a Negative Acknowledgement of XML Receipt message (IE917: C_XML_NCK) containing the unknown namespace error(s). This exception to the normal processing of the message sending/receiving indicates that the XML Schemas of either the sending or the receiving application are not up to date.

The namespaces use the following form:

- for the “IE” Schemas:

urn:publicid:-:EC:DGTAXUD:EMCS:PHASE<phase number>:IE<message number>:V<version>

- for the Helper Schemas:

urn:publicid:-:EC:DGTAXUD:EMCS:PHASE<phase number>:<helper type>:V<version>

where:

- EC: European Commission;
- DG TAXUD: TAXation and Customs Union DG;
- EMCS: Excise Movement and Control System;
- PHASE<phase number>: the number of the phase;
- IE<message number>: the number of the message;
- <helper type>: one of DOC, TCL, TMS, or TYPES;
- V<version>: the version of the Schema version.

IX.I.2 Character set support

XML document and schema always use one encoding, which is specified in the prologue of the document and schema. The encoding supported for all Common Domain message exchanges is the UTF-8 Standard [S3] as also defined in section VIII.I.2.1.2 Character set usage along with the character set supported for all Common Domain message exchanges which is the Unicode Standard [S2].

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section IX - XML formatting	

IX.I.3 XML mapping of Information Exchange

XML mapping of all Information Exchange in EMCS can be found in the corresponding Appendix E.

IX.I.4 XML schema mapping of Information Exchange (XSD)

The XML schemas for all messages used in EMCS are provided in the corresponding directory (Appendix H: Directory with XML Schemas (XSDs)).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Section X TRANSPORT OF MESSAGES VIA CCN/CSI

X.I.1 Introduction

X.I.1.1 Summary

This section specifies the exchange of EMCS messages through the Common Domain.

The specification consists of a set of specified items that are grouped in two parts:

- The mandatory items: these consist of all choices that need agreement in the Common Domain to ensure end-to-end communication between NEAs.
- The recommended items: these consist of all recommendations for the best possible performance of the communication means. The varied architectural constraints that may exist on different NDEA platforms have also been considered.

X.I.1.2 Architectural Assumptions

1. The exchanges in the Common Domain use the communication services provided by the CCN/CSI infrastructure. The CCN/CSI infrastructure has the main objective of exchanging CCN messages across the Common Domain: a CCN message is an object that is created, sent and received through the CSI API. A CCN message has a structure composed of service elements and Application data. The CCN infrastructure consists of a set of interconnected Gateways; each NEA is able to use the CCN services by connecting a NEA front end to a nationally located Gateway and accessing this Gateway through the CSI API.
2. Only the subset of CCN asynchronous communication services is used in EMCS.
3. The CCN asynchronous services operate on persistent storage objects named “queues”. In this mode, CCN messages are exchanged between queues by the CCN services. There exists a consistent naming convention for all queues defined for EMCS.
4. In EMCS, the Application data within a CCN message consists of one Information Exchange.
5. The responsibility for routing a CCN message coming from the Common Domain, once it is present in the receiving queue, entirely lies upon the NEA. This includes the steps of (a) reading from the receiving queue; (b) dispatching the message contents to processes and destinations within the National Domain; (c) extracting where appropriate the XML message from the CCN message; (d) reacting appropriately to the XML message contents in accordance with the EMCS Business Process threads, rules and conditions.
6. The Information Exchanges do not require an immediate answer. However, the duration between a request and an answer (functional IE) is at maximum 15 minutes.
7. A NEA is interacting with CCN/CSI via CCN/CSI API calls. It is mandatory to check the result of any API call (by checking the API return and reason codes) to assure the proper execution of the API call and to take corrective actions in case of problems.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

8. Whenever an Information Exchange is sent, CCN/CSI enables to request report messages indicating the state of the message transfer. The usage of these report messages is mandatory. The sender must check the reception of the report messages and corrective action needs to be taken in case of problems.
9. CCN/CSI enables automated ARC nursing, whereby an automated application⁹ can track all CCN/CSI exchanges related to a single ARC and sequence number, including the exchange of the various CCN/CSI reports. In order to support ARC nursing, it is highly recommended to follow the conventions defined for the usage of CorrelId in the CCN/CSI exchanges.
10. In EMCS, when two Information Exchanges are sent to the same queue on the same Gateway and the sequence of these is significant then it is the responsibility of the sending NEA to ensure that this sequence is maintained under all conditions. This may be achieved by delaying the sending of the second until receipt of the first has been acknowledged.
11. In order to prevent a message from being deleted from a queue before it has actually been processed, the usage of the verb `get` is not allowed since it will effectively delete the message from the queue. Instead, the queue should first be browsed using the verb `browse`; if the retrieval and processing of the message is successful it may be deleted from the queue using the verb `delete`.

X.I.2 Topology

X.I.2.1 References to CCN/CSI

The exchange in the Common Domain may use the communication services provided by the CCN/CSI infrastructure.

This document does not describe CCN/CSI. That information is available in the relevant CCN/CSI manuals. The document provides a brief overview of the CCN communication, with the aim of helping the reader to find his/her way into the CCN/CSI documentation and it also defines the programming choices applicable to EMCS.

The definitions of data structures, of data types and of constants are found in “include” files (for the C language) or “COPY” files (for the COBOL language) and or “JCSI (for the Java language). “Library” files providing the compiled code that must be linked with the application-compiled code are also provided. The indications of the files to use are given in:

- For the C language: see [A3] “Building an application on UNIX systems”.
- For the COBOL language on BS2000: see [A4].
- For the COBOL language on CICS: see [A5]

⁹ Currently, the application tracking CCS/CSI messages is CS/MISE (see Sub-Section V.II CS/MISE) through the ARC follow-up functionality.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- JCSI Reference Manual for the Java language;

X.I.2.2 The CCN communication reminder

This chapter presents those elements of CCN/CSI that are agreed to ensure end-to-end communication between two CCN Gateways.

In this chapter, the word “message” is to be understood as “CCN message”.

CCN carries messages between Gateways. The messages are prepared and sent by a sending Application, using CSI API; the messages are received and interpreted by a receiving Application. The communication uses the asynchronous mode.

An application is said to communicate in asynchronous mode when this application is able to send a message without having established a connection with its peer application. Messages are exchanged by placing them in and extracting them from, queues.

Figures available from an application currently operational on CCN/CSI demonstrate that the total request/reply round trip delay - i.e. until to receive the Confirmation on Arrival (CoA) - is in the range of 5 to 10 seconds with all accesses performed in asynchronous mode. This is the time between the moment that the sender puts a Request on its sending queue and the moment when the reply is available for the sender to retrieve from its receiving queue.

The structure of the message consists of:

- A description of the message or “**message descriptor**” (see X.I.2.2.1).
- A description of the data or “**data descriptor**” (see X.I.2.2.2); the data descriptor is said to ‘contain’ the data, even when actually it contains only a description consisting of length (in bytes) and address in memory or location in the file system.
- A description of specific parameters, called “**quality of services**”. These parameters describe particular handling that has to be applied (when sending) or was applied (when receiving), on the data contained in the data descriptor during execution of a CSI verb.

These three structure components are further detailed in the three paragraphs (X.I.2.2.1, X.I.2.2.2 and X.I.2.2.6) that follow.

The sending and receiving of CCN messages may occur after the application has connected itself to the CCN Gateway, has created a security context and has connected to the queue manager: these steps are further detailed in the paragraphs (X.I.2.2.8, X.I.2.2.9, X.I.2.2.10 and X.I.2.2.11) that follow.

A NEA will always interact with CCN/CSI via CCN/CSI API calls. It is essential that the correct execution be checked after each API call (by checking the API return code) and that appropriate action is taken when the API call has failed. Possible API return codes are documented in [A9].

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.2.2.1 The message descriptor

The message descriptor consists of a CSIMQMD structure.

This structure is to be prepared, prior to sending a message with the CSI_mq_put verb.

This structure is to be interpreted and some of its elements have to be copied back in an equivalent structure when an Information Exchange is replied with another Information Exchange, in line with the Time Sequence Diagram demonstrated in paragraph X.I.2.2.7.

Table 39 details the CSIMQMD structure members.

Typedef struct tag		Value on SEND	Value for CCN Report	Notes
CSIMQMD {				
CSICHAR4	StrucId;	CSIMQMD_STRUC_ID	CSIMQMD_STRUC_ID	
CSILONG	Version;	CSIMQMD_VERSION_1	CSIMQMD_VERSION_1	
CSILONG	Report;	0L	0L	
CSILONG	MsgType;	CSIMQMT_REQUEST or CSIMQMT_DATAGRAM	CSIMQMT_REPORT	
CSILONG	Expiry;	3 456 000L	(DNC)	
CSILONG	Feedback;	CSIMQFB_NONE		
CSILONG	Encoding;	0L	(DNC)	
CSILONG	CodedCharSetId;	0L	(DNC)	
CSICHAR8	Format;	Empty string	(DNC)	
CSILONG	Priority;	0L	(DNC)	
CSILONG	Persistence;	CSIMQPER_PERSISTENT	(DNC)	Message persistence
CSIBYTE24	MsgId;	CSIMQCI_NONE	= MsgId of reported Msg	Message identifier
CSIBYTE24	CorrelId;	= ARC + sequence number	= CorrelId of reported Msg	
CSILONG	BackoutCount;	0L	(DNC)	Backout counter
CSICHAR48	ReplyToQ;	Empty string	(DNC)	Not used
CSICHAR48	ReplyToQMgr;	Empty string	(DNC)	Not used
CSICHAR12	UserIdentifier;	Empty string	(DNC)	Not used
CSICHAR32	AccountingToken;	Empty string	(DNC)	Not used
CSICHAR32	AppIdentityData;	Empty string	(DNC)	Not used
CSILONG	PutApplType;	0L	(DNC)	Not used
CSICHAR28	PutApplName;	Empty string	(DNC)	Not used
CSICHAR8	PutDate;	Empty string	(DNC)	Not used
CSICHAR8	PutTime;	Empty string	(DNC)	Not used
CSICHAR4	ApplOriginData;	Empty string	(DNC)	Not used
} CSIMQMD;				

Table 39: MQ Message Descriptor

Notes:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

1. Column “Value on SEND” exhibits the value that an application has to set in each structure member.
2. Column “Value for CCN report” defines the value set in the CCN reports.
3. The indication “(DNC)” means: “Do not consider”.
4. Values in uppercase are CCN/CSI constants defined in the header files.
5. The value CSIMQMT_REQUEST has the effect of requesting a reply. The name of the queue to which the reply is to be sent is contained in ReplyToQ field of CSIQOS structure.
6. The CCN Gateway buffers incoming messages in case the application is temporarily unable to process them. The requirement is to be able to buffer the messages for 96 hours (in tenths of second, this duration amounts to: $96 \times 3600 \times 10 = 3\,456\,000$).
7. Value in ‘Feedback’ is set by the queue manager to indicate, within a report message, how the original message was handled on the Destination queue. The reports are sent back to the sender as specified by the parameters set in the configuration (Section X - Transport of Messages via CCN/CSI). Possible values are:
 - A value comprised in the range from *Exception report*
CSIMQFB_SYSTEM_FIRST and
CSIMQFB_SYSTEM_LAST (those limits
included)
 - CSIMQFB_EXPIRATION *Expiration report*
 - CSIMQFB_COA *Confirm on Arrival report*
 - CSIMQFB_COD *Confirm on Delivery report*
8. The MsgId value is an identifier used by the application to correlate a Report Message with the Information Exchange it reports about. As an Expiration report may only be generated after 96 hours (see Note 2 for ‘Expiry’ field above), it is recommended that the MsgId generating rule uses a counter that does not “rewind” in less than 96 hours.

As the field MsgId presents 24 bytes, the NDEA designer is able to choose a MsgId definition that covers this condition and well beyond.

For the generation of the MsgId field the following conventions should be followed:

- When sending a message, the MsgId is generated from the sending application.
- When a report is sent back, the MsgId is equal to the MsgId of the original message.

Therefore, the MsgId needs to be set to CSIMQMI_NONE. The queue manager will then generate a unique message identifier upon sending.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

The MsgId of the original message is copied into the MsgId of the report message by setting the appropriate flag CSIMRQRO_PASS_MSG_ID in the ReportRequest field of the QOS.

9. For the generation of the CorrelId field the following conventions should be followed (the decision process is presented as a flowchart in Figure 225):

- The CorrelId field of the report messages should be equal to the CorrelId of the original message by setting the appropriate flag CSIMRQRO_PASS_CORREL_ID in the ReportRequest field of the QOS. A report message is sent back to the sender:
 - Confirm on Delivery (CoD) report when the message has been read by the receiving application and deleted from the queue.
 - Confirm on Arrival (CoA) report when the message has arrived on the remote Gateway.
 - Expiration (EXP) report when a value of time lapse set in the CSIMQMD.Expiry variable (see Note 2) has expired and an application tries to retrieve this message after the elapsed expiration time: the message, once arrived on Destination queue (CoA), was not fetched from this queue by an application program during the time allotted.
 - Exception (EXC) report when other technical errors that are related to the queuing system have occurred.
- The usage of the four report messages (CoA, CoD, EXP, EXC) is mandatory. A NEA will have to request all four types of reports whenever an Information Exchange is sent and will have to wait for the reception of the reports for any Information Exchange sent. In case of problems, corrective action needs to be taken. This usage is further discussed in paragraph X.I.2.2.7.
- The report messages are to be read from the queue whose name is given by the element “ReplyToQ” of the Quality of Service structure. See Table 42. The report messages do not contain the original message.
- In case the report message could not be delivered to the queue indicated by this element, it will be stored on the dead-letter queue (CSIMQRO_DEAD_LETTER_Q) on the gateway that sent the report. Each CCN Gateway needs to have this type of queue.
- The CorrelId field of the IE702, IE714, IE906, and IE917 refusal messages should be equal to the value of the MsgId field of the corresponding erroneous message (IE702 for an erroneous IE701, IE714 for an erroneous IE713, and IE906 and IE917 for any erroneous message). This will enable the correlation of refusal messages to the corresponding erroneous messages.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- In order to support ARC nursing, the CorrelId field of the messages exchanged concerning a movement (IE717, IE801, IE802, IE803, IE807, IE810, IE813, IE818, IE819, IE837, IE840, IE861, IE871, IE904, IE905, and IE934) should be equal to the ARC + Sequence Number (in an ASCII format).
- The CorrelId field of the rest of the messages in EMCS should be equal to the empty string.

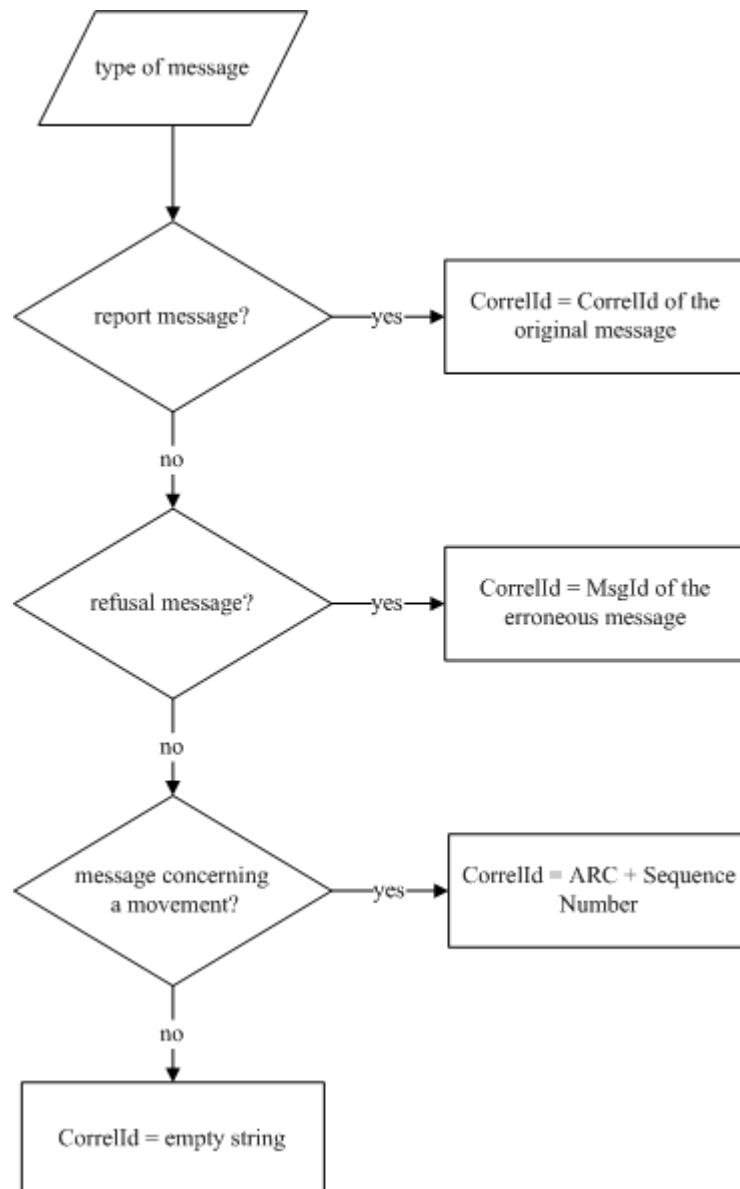


Figure 225: The generation of the CorrelId field value decision process

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.2.2.2 The data descriptor

The data descriptor is implemented by a CSIDD structure shown in Table 40.

Typedef struct tag		Value on SEND	Data descriptor
CSIDD {			
CSICHAR4	StrucId;	CSIDD_STRUC_ID	Structure identifier
CSILONG	Version;	CSIDD_VERSION_1	Structure version
CSILONG	Flags;	O_MEMORY or O_FILE	
CSICHAR256	FileName;	See X.I.2.2.4	
CSIULONG	DataLength;	See X.I.2.2.4	
CSIBYTE	*Data;	See X.I.2.2.4	
} CSIDD;			

Table 40: CSI Data Descriptor

Notes:

1. Column “Value on SEND” exhibits the value that an application has to set in each structure member.
2. The CSIDD structure allows the representation of data:
 - Either located in core memory: the structure element ‘Flags’ must have the value O_MEMORY.
 - Or located in a file: the structure element ‘Flags’ must have the value O_FILE.

The choice of which method to use for passing the application data is strongly dependent on the design choices taken for the application: it is recommended that the “O_MEMORY” method be used whenever possible.

X.I.2.2.3 Allocation of a CSIDD

The CSIDD has to be allocated dynamically and initialised with the function HL_alloc(), documented in [A8]:

```
CSILONG HL_alloc(
    CSIULONG SizeRq,
    CSIDD **DataOut,
    CSILONG*ReturnCode,
    CSILONG *ReasonCode
);
```

If the data are represented in core memory, the value of argument SizeRq must be at least equal to or greater than, the length of the application data (expressed in octets).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Upon successful call to HL_alloc(), some elements of the allocated CSIDD structure are initialised:

- StrucId = CSIDD_STRUC_ID.
- Version = CSIDD_VERSION_1.
- Flags = O_MEMORY.
- DataLength = 0L.

When a CSIDD structure is not used anymore, it has to be given back to system resources by performing the HL_free() verb, documented in [A8]:

```
CSILONG HL_free(
    CSIDD **DataOut,
    CSILONG*ReturnCode,
    CSILONG *ReasonCode
);
```

The example for allocation of a CSIDD structure is part of the listing in [R5].

X.I.2.2.4 Inserting the application data into the CSIDD structure

The application data consist of an Information Exchange in XML format. In the example provided in Figure 226, they are represented as a core memory value named 'Request'.

- If the application data are represented in core memory:
 - They have to be copied from their current location into the buffer allocated and pointed to by CSIDD.Data and the element "DataLength" must be entered as the exact length (expressed in octets) of the application data. It cannot be assumed that the Information Exchange is represented in memory as a NULL-terminated string according to the ANSI C convention. The example for this case is provided (as statement "memcpy").
- If the application data are represented in a file:
 - The elements of the CSIDD structure must be set to:
 - Flags = O_FILE.
 - DataLength = length of useful contents in the File.
 - FileName = full path name of the file. 'FileName' is set to the path name of a file containing application data.

X.I.2.2.5 Encoding the CSIDD for EMCS

When the CSIDD is prepared, with elements either FileName or Data (as well as DataLength) correctly initialised by the application, it has to be presented, on the sending side to the HL_encode() verb. Conversely, after being received, a CSIDD structure must be presented to the HL_decode().

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

The prototype of the HL_encode verb is:

```

CSILONG HL_encode(
  IN CSIDD      *DataIn,
  IN CSICHAR32  MsgTypeId,
  IN CSILONG    CodePage,
  IN CSICHAR128 HostFormat,
  OUT CSIDD     *DataOut,
  OUT CSILONG   *ReturnCode,
  OUT CSILONG   *ReasonCode
);

```

The parameter MsgTypeId must have a value within a pre-determined set of values that are listed in Table 40 under Header MsgTypeId. These values are actually produced by the CCN-TC as the result of the operations described in Table 41.

IE	IE Name	Reference	Message Type string
701	Common Request	C_REQ_SUB	"CD701A-MSG.emcs"
702	Refusal of Common Request	C_REQ_REF	"CD702A-MSG.emcs"
713	Incremental Update or Full Register of Economic Operators	C_QRO_DAT	"CD713A-MSG.emcs"
714	Refusal of Update of Economic Operators	C_QRO_REF	"CD714A-MSG.emcs"
721	Administrative Cooperation Common Request	C_COO_SUB	"CD721A-MSG.emcs"
734	Reference Data Dissemination	C_RDD_REF	"CD734A-MSG.emcs"
742	SEED Statistics	C_STA_VAL	"CD742A-MSG.emcs"
770	Business Process Unavailability Schedule	C_AVA_DAT	"CD770A-MSG.emcs"
801	E-AD	C_EAD_VAL	"CD801A-MSG.emcs"
802	Reminder Message for Excise Movement	C_EXC_REM	"CD802A-MSG.emcs"
803	Notification of Diverted e-AD	C_EAD_NOT	"CD803A-MSG.emcs"
807	Interruption of Movement	C_STP_NOT	"CD807A-MSG.emcs"
810	Cancellation of e-AD	C_CAN_DAT	"CD810A-MSG.emcs"
813	Change of Destination	C_UPD_DAT	"CD813A-MSG.emcs"
818	Accepted or Rejected Report of Receipt/Export	C_DEL_DAT	"CD818A-MSG.emcs"
819	Alert or Rejection of an e-AD	C_REJ_DAT	"CD819A-MSG.emcs"
820	History Results	C_HIM_RES	"CD820A-MSG.emcs"
821	List of e-AD as Result of a General Query	C_LST_VAL	"CD821A-MSG.emcs"
829	Notification of Accepted Export	C_EXP_NOT	"CD829A-MSG.emcs"
837	Explanation on Delay for Delivery	C_DEL_EXP	"CD837A-MSG.emcs"
839	Customs Rejection of e-AD	C_CUS_REJ	"CD839A-MSG.emcs"
840	Event Report	C_EVT_DAT	"CD840A-MSG.emcs"
861	Basis for Recovery of Duties	C_RES_DAT	"CD861A-MSG.emcs"
867	Administrative Cooperation Results	C_COO_RES	"CD867A-MSG.emcs"

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

IE	IE Name	Reference	Message Type string
868	Answer Message	C_COO_ANS	"CD868A-MSG.emcs"
869	Reminder Message for Administrative Cooperation	C_COO_REM	"CD869A-MSG.emcs"
871	Explanation on Reason for Shortage	C_SHR_EXP	"CD871A-MSG.emcs"
904	Status Request	C_STD_REQ	"CD904A-MSG.emcs"
905	Status Response	C_STD_RSP	"CD905A-MSG.emcs"
934	Data Packaging	C_PAC_DAT	"CD934A-MSG.emcs"
906	Functional NACK	C_FUN_NCK	"CD906A-MSG.emcs"
917	Negative Acknowledgement of XML receipt	C_XML_NAC	"CD917A-MSG.emcs"

Table 41: MsgTypId used for an Information Exchange of EMCS

See related chapter X.I.5.2 for an explanation of arguments 'CodePage' and 'HostFormat' as obtained from configuration information.

The listing in R5 illustrates the usage of HL_alloc and HL_encode verbs, as well as the initialisation of the CSIDD structure with the XML Information Exchange.

```

REQUESTrequest;
CSIDD pCSirequest;
CSILONGReturnCode;
CSILONGReasonCode;

The variable Request has been prepared in REQUEST structure. Assume
Request is the XML _interchange for a CD801A message

Steps are:
(1) (see X.I.2.2.3) Allocate Data Descriptors for request
HL_alloc (sizeof (REQUEST), &pCSirequest, &ReturnCode, &ReasonCode);

(2) (see 2.1.2.2) Copy REQUEST structure into "Data" field of request
CSIDD structure
memcpy (pCSirequest->Data (CSICHAR *) &Request, sizeof (REQUEST));

pCSirequest->DataLen = sizeof (REQUEST);

(3) (see 2.1.2.3) Encode request
HL_encode (pCSirequest, "CD801A-MSG.EMCS", CODEPAGE, HFMT,
pCSirequest, &ReturnCode, &ReasonCode);

After these steps: the CSIDD will be sent with HL_mq_put() verb.
Logging of this CSI movement will take a record of "CD801A"

```

Figure 226: Example of CSIDD allocation, initialisation with Information Exchange and encoding

This code is not intended to be used directly as actual working code. All API return codes (and possibly reason codes) will have to be checked after every API call.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.2.2.6 The quality of service

CCN/CSI defines a structure - called CSIQOS - to specify (or to retrieve) which type of particular handling has to be applied (or has been applied) on the data contained in a given Data Descriptor parameter of a verb during execution of this verb.

A default QoS is pre-defined for an application via a configuration parameter. This default QoS is retrieved by the CSI stack during the session establishment. This QoS structure should be used by the programmer. However, two fields of the QoS are not defined in the configuration and have to be specified for each call: the priority and the report reply queue.

Requirements for the message descriptor:

- This structure is to be prepared, prior to sending a message.

The Quality of Service is represented by the CSIQOS structure shown in Table 42. The fields applicable in this structure are ticked with a “✓” in the right-hand column of this Table.

typedef struct tag		value on SEND	Notes	
CSIQOS{				
CSICHAR4	StrucId;	CSIQOS_STRUC_ID	Structure id.	✓
CSILONG	Version;	CSIQOS_VERSION_1	Structure version	✓
CSILONG	QoSToApply;		Specified QoS	✓
CSILONG	AppliedQoS;	(DNC)	Applied QoS	✓
CSILONG	Priority;		Priority value	✓
CSILONG	ReportRequest;		Report/Request flag	✓
CSICHAR48	ReplyToQ;		Name of reply queue	✓
CSICHAR48	ReplyToQMgr;	(DNC)	Name of reply queue mgr	✓
CSIBYTE24	CorrelId;	Empty string	Correlation id.	✓
CSILONG	Integrity;		Integrity flag	✓
CSILONG	Confidentiality;		Confidentiality flag	✓
CSILONG	Compression;		Compression flag	✓
CSICHAR8	CompressionId;		Comp. Algorithm id.	✓
CSICHAR16	CoT;	DEFAULTCOT	Class of Traffic	✓
CSICHAR48	VASScript;	Empty string	VAS script name	✓
CSILONG	DegradedMode;	(DNC)	Degraded mode flag	
} CSIQOS;				

Table 42: CCN/CSI Quality of Service structure

Notes:

1. Column “Value on SEND” exhibits the value that an application has to set in each structure member.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

2. ‘QoSToApply’ is a vector of bits obtained by adding once or by performing a bitwise OR operation between, the set of values that follow:

QoSToApply =

CSI_QOS_PRIORITY +

CSI_QOS_REPLYTO_Q +

(0 or CSI_QOS_INTEGRITY: see Note 5.a) +

(0 or CSI_QOS_CONFIDENTIALITY: see Note 5.b) +

(0 or CSI_QOS_COMPRESSION: see Note 5.c) +

(0 or CSI_QOS_COMPRESSION_ID: see Note 5.c)

Two priority values are distinguished (these values are written with the C language convention for long integers):

- a) High priority: QoS priority parameter has value 7L.
- b) Normal priority: QoS priority parameter has value 5L. Normal priority is applicable to the other Information Exchanges exchanged across the Common Domain.

When a message is retrieved from a Destination queue, it is accompanied with the QoS that was set at the time of sending. Therefore, the priority set when sending has to be used to fetch the received message in accordance with the rule:

Rule for fetching messages from a receiving queue:

High priority messages will always be processed first and within the same priority level a “first in, first out” behaviour will be used.

3. ‘ReportRequest’ specification is fully defined by configuration (see chapter X.I.5) and hence is not to be set for each individual message. One thus has the choice to use either the default value set by configuration or to use an individual value for every individual message sent. In any case it is required to request all 4 types of reports (CoA, CoD, EXC, EXP) whenever an Information Exchange is sent. In addition, the following flags need to be set for copying MsgId and CorrelId from the original message to the report message:
 - C_SIMQRO_PASS_MSG_ID
 - C_SIMQRO_PASS_CORREL_ID
4. ‘ReplyToQ’ is to be set to the name of a queue that exists on the CCN Gateway used by the sending application. This name is of no use for the receiving application (because the report message is constructed and sent by the CCN software and not by the receiving application).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

5. The attributes “Integrity”, “Confidentiality” and “Compression” apply to the link between the NDEA platform and the CCN gateway [R3] and [A8]. Therefore, the choice of modifying the values preset at Configuration time (see paragraph X.I.5.2.1) depends of the strength of this link as perceived by the NDEA Local Security Officer.

The following notes apply to the situation where the decision is taken to change the default values.

a) If the Nationally Developed Excise Application chooses to set the Integrity for a specific message differently from the configured value, then

- Value CSI_QOS_INTEGRITY must be added to QoSToApply
- Element ‘Integrity’ in CSIQOS takes the value CSI_INTEGRITY_REQUIRED.

Else:

- Value CSI_QOS_INTEGRITY must not be added to QoSToApply.
- Element ‘Integrity’ in CSIQOS takes the value CSI_INTEGRITY_NOT_REQUIRED

b) See chapter X.I.5 for the explanation of the configuration options set for the Confidentiality (data are encrypted). If the National Developed Application or the NEA chooses to set the Confidentiality for a specific message differently from the configured value, then:

- Value CSI_QOS_CONFIDENTIALITY must be added to QoSToApply (see Note 2).
- Element ‘Confidentiality’ in CSIQOS takes the value CSI_CONFIDENTIALITY_REQUIRED.

Else:

- Value CSI_QOS_CONFIDENTIALITY must not be added to QoSToApply (see Note 2).
- Element ‘Confidentiality’ in CSIQOS takes the value CSI_CONFIDENTIALITY_NOT_REQUIRED.

c) See chapter X.I.5 for the explanation of the configuration options set for the Compression and CompressionId. If the Nationally Developed Excise Application chooses to set these parameters for a specific message differently from the configured value, then:

- Value CSI_QOS_COMPRESSION and CSI_QOS_COMPRESSION_ID must be added to QoSToApply (see Note 2).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- Element 'Compression' in CSIQOS takes the value CSI_COMPRESSION_REQUIRED.
- Element 'CompressionId' in CSIQOS takes the value CSI_COMPRESSIONID_LZW.

Else:

- Value CSI_QOS_CONFIDENTIALITY must not be added to QoSToApply (see Note 2).
- Element 'Confidentiality' in CSIQOS takes the value CSI_COMPRESSION_NOT_REQUIRED.
- Element 'CompressionId' in CSIQOS takes the value CSI_COMPRESSIONID_NONE.

X.I.2.2.7 Illustration of the use of the QoS parameters

This section illustrates by means of Time Sequence Diagrams the use of the QoS parameters and of the CCN report messages. It shows the communication between a sending CSI stack, a sending CCN Gateway, a receiving CCN Gateway and a receiving CSI stack. It shows several cases starting with the normal case in which a Confirm on Arrival and Confirm on Delivery are received. Based on this normal case, several exceptions are shown in Figure 227.

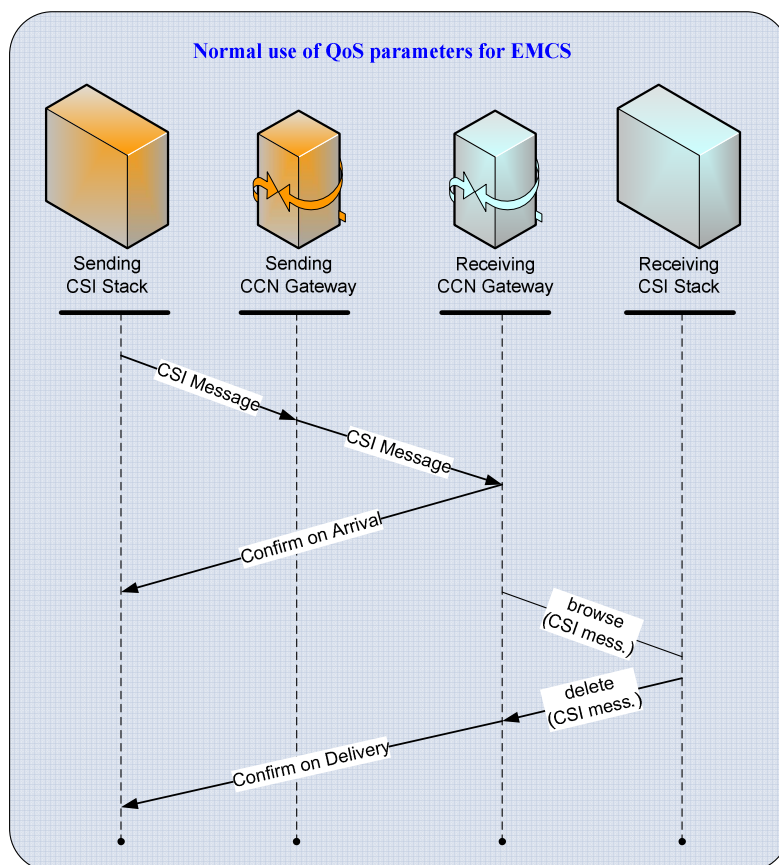


Figure 227: Normal use of QoS parameters for EMCS

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Figure 227 shows that a queue is to be first browsed for a message, before the message is deleted (after having processed the message). This use of CSI verbs is mandatory, in order to prevent a message from being deleted from a queue before it has actually been processed. The usage of the verb HL_mq_get is not allowed.

It should be noted that, for every Information Exchange sent, the user will have to wait for the reception of the CoA and CoD before concluding that the message has been transferred successfully. The CoA is denoting that the Information Exchange has reached the destination queue, while the CoD is denoting that the Information Exchange has been successfully processed at (and deleted from) the destination queue. Please also note that CCN/CSI does not assure the delivery of CoA and CoD in sequence (in rare occasions, the CoD may be returned first).

If a message cannot be sent to a Destination CCN Gateway, the result is indicated in the CSI verb (HL_mq_put). This type of error needs to be handled by the sending CSI stack. In such a case, the CSI message will never arrive at its Destination CCN Gateway and no CCN report message will be generated.

An exception report can be generated for various reasons and can be generated by both sending and receiving Gateways. Whenever a sending application is submitting a message to CCN/CSI via the HL_mq_put verb, the message will first be stored in the internal technical queues of the CCN/CSI stack on the sending Gateway. CCN/CSI will then attempt to forward the message from the internal technical queues on the sending Gateway to the internal technical queues on the receiving Gateway and from these to the destination queue on the destination Gateway. An exception report is generated whenever an anomaly is detected in the internal CCN/CSI behaviour (at either sending or receiving Gateway). This can e.g. be:

- Incorrect addressing at CCN/CSI level, either at sending or receiving side (e.g. invalid Gateway or invalid queue name)
- Unavailability of the receiving Gateway when attempting to transfer the message.

The CCN/TC is maintaining a number of availability flags for the different Gateways. Whenever a Gateway is marked, as down, an EXC report will immediately be generated upon sending to this Gateway. When the Gateway is not marked, as down, CCN/CSI will attempt to transfer the message; if this does not succeed, an EXC report can still be generated by the sending Gateway.

As Figure 228 depicts, the Expiration report can be created when the message has arrived at the destination Gateway (Confirm on Arrival has been created) and when the original message has not been read from the Destination queue before the timer set by the 'Expiry' field of the message descriptor expires. The Destination CCN Gateway handles the expiration timer. The destination timer is only checked whenever the destination queue is accessed (no expiration reports will be sent when the queue is not accessed at all). Therefore, EXP reports will not always be generated when the expiry took place (they may actually be generated afterwards).

For all these reasons it is therefore mandatory to wait for the processing reports after sending an Information Exchange and to maintain a timer for every Information Exchange sent. When this timer expires, one should check the availability of the destination Gateway (and possibly re-send the message).

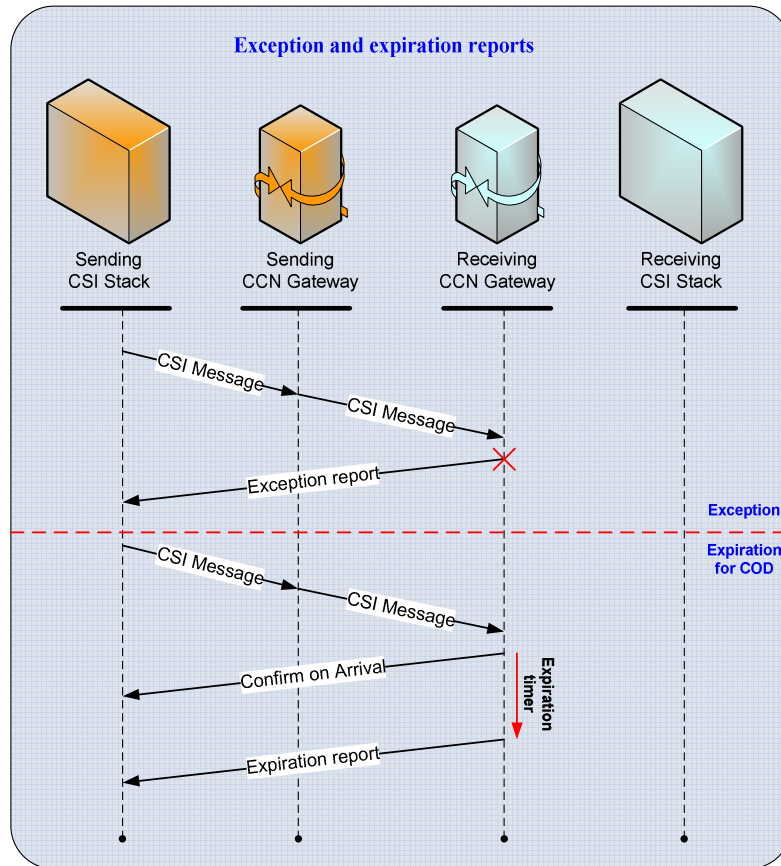


Figure 228: Exception and expiration reports

All possible options for the use of the QoS parameters and their exceptions are shown in the State Transition Diagram in Figure 229. This State Transition Diagram specifies the states of one CSI message present in the sending CSI stack, with respect to the use of CCN. It assumes that the binding of the CSI stack to the CCN Gateway has successfully taken place.

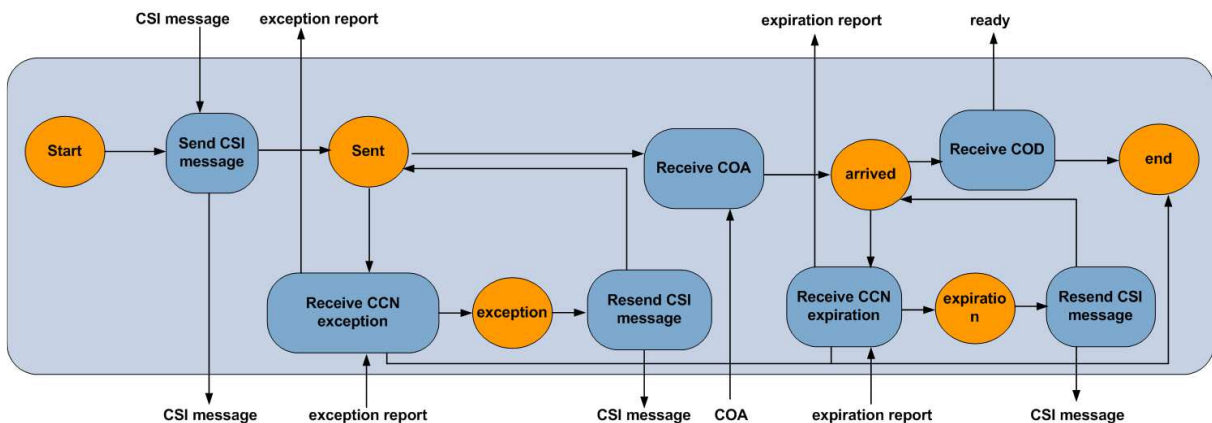


Figure 229: State Transition Diagram of the sending CSI stack (normal flow)

Each of the transitions shown in the previous figure consists of CSI verbs.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.2.2.8 Connecting the application to the CCN Gateway

Any instance of any application establishes a CSI session by using the HL_bind() verb:

```
CSILONG HL_bind(
    CSICHAR48    AppliName,
    CSICHAR48    ProxyName,
    CSIQOS       *DefaultQoS,
    CSILONG      *ReturnCode,
    CSILONG      *ReasonCode
);
```

“AppliName” is a value previously registered to the CCN Gateway. A remote proxy name must have been set together with the ApplicationName at configuration time. This default value may not be overridden: therefore the “ProxyName” argument in the HL_bind() call must be an empty string.

Closing a CSI session occurs with the verb HL_unbind().

Note that the session, when established, has no identifier on its own.

X.I.2.2.9 Creating a security context for an application

Within the session, a security context has to be created by using the verb HL_init_sec_context and is destroyed with the verb HL_delete_sec_context. The parameter of the HL_init_sec_context verb is a CSISECINFO containing (in the current version of CSI software) a GSSNAME structure:

```
typedef struct tagGSSNAME {
    char    user_id[GSS_NAME_LENGTH];
    char    application_id[GSS_NAME_LENGTH];
    char    user_password[GSS_PASSWD_LENGTH];
    char    application_key[GSS_PASSWD_LENGTH];
} GSSNAME;
```

The components of the GSSNAME structure are:

- A user identification.
- An application name.
- A user password.
- An application key.

These four values must have been configured previously by the NDEA Local Security Officer.

The structure GSSNAME is referenced (i.e. pointed) by a structure CSISECINFO:

```
typedef struct tagCSISECINFO {
    CSILONG securityType;
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

CSIVOID securityInfoP;    / here: GSSNAME */
} CSISECINFO;

```

‘securityType’ is set to the value BI_SEC_TYPE.

Finally the verb HL_init_sec_context() is used with the structure CSISECINFO that was initialised:

```

CSILONG HL_init_sec_context(
    CSISECINFO *credentialInfo,
    CSILONG    *ReturnCode,
    CSILONG    *ReasonCode
);

```

Note that the security context, when established, has no identifier on its own.

The verb HL_delete_sec_context() is used for deleting the security context.

X.I.2.2.10 Connecting to the queue manager

The application has to connect itself to the queue manager in order to be able to issue commands related to queue access. The verb HL_mq_conn() is used for this purpose:

```

CSILONG HL_mq_conn(
    CSICHAR48    Name,
    CSIMQHCONN  *Conn,
    CSILONG     *ReturnCode,
    CSILONG     *ReasonCode
);

```

“Name” value must be set to an empty string for current version of CSI.

“Conn” is initialised by this function and has to be used subsequently for opening any queue.

The verb HL_mq_disc() is used to disconnect from the queue manager:

```

CSILONG HL_mq_disc(
    CSIMQHCONN  *Conn,
    CSILONG     *ReturnCode,
    CSILONG     *ReasonCode
);

```

Argument “Conn” in this call must be identical to the one obtained from previously using HL_mq_conn().

X.I.2.2.11 Opening a queue

The application has to open a queue before performing any access to it. See in Table 44 the list of verbs that may be used once a queue has been successfully opened.

The verb HL_mq_open() is used to open a queue:

```

CSILONG HL_mq_open(

```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

CSIMQHCONN Conn,
CSIMQOD *ObjDesc,
CSILONG Options,
CSIMQHOBJ *Obj,
CSILONG *ReturnCode,
CSILONG *ReasonCode
);

```

Value of argument “Conn” is identical to the one obtained from HL_mq_conn().

Value of argument “Options” is obtained by adding values that control the HL_mq_open() behaviour. There is no mandatory value of this argument in DDNEA for Phase 3 because the policy for reading and writing into Gateway queues by the application is a local matter. However, the recommended use of the verb HL_mq_browse() for retrieving messages from a queue (see X.I.2.2.12.4), as well as the use of default configured values, mandates the use of at least the options CSIMQOO_BROWSE and CSIMQOO_INPUT_AS_Q_DEF. Hence:

Options = CSIMQOO_BROWSE + CSIMQOO_INPUT_AS_Q_DEF.

Successful call of the verb HL_mq_open() for a certain queue provides a value “Obj”, called a “handle”, that will be subsequently used in all verbs (listed in Table 44) that deal with this queue.

Argument “ObjDesc” is a CSIMQOD structure that must be initialised as follows:

Typedef	struct tag	Initial value	Notes
CSIMQOD {			
CSICHAR4	StrucId;	CSIMQOD_STRUC_ID	Structure id.
CSILONG	Version;	CSIMQOD_VERSION_1	Structure version
CSILONG	ObjectType;	CSIMQOT_Q	
CSICHAR48	ObjectName;		Name of queue
CSICHAR48	ObjectQMgrName;	(DNC)	This field must not be used
CSICHAR48	DynamicQName;	(DNC)	This field must not be used
CSIBYTE12	AlternateUserId	(DNC)	This field must not be used
} CSIMQOD;			

Table 43: MQ Object Descriptor

Notes

The Queue Name is inserted here. This allows a maximum length of 47 characters for the Queue Name.

When not used anymore, a queue must be closed by using the verb CSI_mq_close:

```

CSILONG HL_mq_close(
CSIMQHCONN Conn,
CSIMQHOBJ *Obj,
CSILONG Options,
);

```


EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

CSILONG *ReturnCode,
CSILONG *ReasonCode
);

```

Value of argument “Conn” is identical to the one obtained from HL_mq_conn().

Value of argument “Obj” is identical to the one obtained from HL_mq_open().

Value of argument “Options” must be set to CSIMQCO_NONE.

X.I.2.2.12 CSI verbs allowed for queue accesses

When a queue was successfully opened, it is identified by a handle Obj, of type CSIMQHOBJ.

The following verbs may be used to deal with the “Obj” queue contents:

Class of verbs	Verb	Usage
close queue	HL_mq_close()	Mandatory.
write in queue	HL_mq_put()	Mandatory (see also X.I.2.2.12.2 however).
read from queue	HL_mq_browse	Mandatory
delete from queue	HL_mq_delete	Mandatory. A message must have been successfully read and processed before deleting.

Table 44: CSI verbs for queue access

It is the responsibility of the application to organise the reading from a receiving queue in order to avoid loss of messages.

X.I.2.2.12.1 Putting a message into a queue: HL_mq_put()

```

CSILONG HL_mq_put(
CSIMQHCONN Conn,
CSIMQHOBJ ObjDesc,
CSIMQMD *MsgDesc,
CSIMQPMO *PutMsgOpts,
CSIDD *DataIn,
CSIQOS *QoS,
CSILONG *ReturnCode,
CSILONG *ReasonCode
);

```

Value of argument “Conn” is identical to the one obtained from HL_mq_conn().

Value of argument “Obj” is identical to the one obtained from HL_mq_open().

Argument “MsgDesc” points to a CSIMQMD structure that was prepared as explained in X.I.2.2.1 The values present in this structure after a successful call of the HL_mq_put() are not to be considered.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Argument “DataIn” points to a CSIDD structure that was prepared as explained in Section X.I.2.2.5.

Argument “QoS” points to a CSIQOS structure that was prepared as explained in Section X.I.2.2.12.2.

Argument “PutMsgOpts” is to be initialised with the statements:

- a) Define a static variable s_DefMQPMO of type CSIMQPMO initialised with constant values:

```
CSIMQPMO s_DefMQPMO = {CSIMQPMO_DEFAULT};
```

- b) Each time the HL_mq_put() verb will be used, define and initialise dynamically a variable s_MQPMO of type CSIMQPMO by using the static variable s_DefMQPMO:

```
CSIMQPMO s_MQPMO;
memcpy(&s_MQPMO, s_DefMQPMO, sizeof(CSIMQPMO));
```

X.I.2.2.12.2 Putting a message into a queue: HL_mq_put1()

The verb HL_mq_put1() is identical to the verb HL_mq_put(), with the exception that it uses an additional argument to specify the queue that has to be written to, instead of a queue handle. This verb performs in one call the function of the three operations: HL_mq_open(), HL_mq_put(), HL_mq_close().

X.I.2.2.12.3 Reading a message from a queue: HL_mq_get()

```
CSILONG HL_mq_get(
    CSIMQHCONN Conn,
    CSIMQHOBJ ObjDesc,
    CSIMQMD *MsgDesc,
    CSIMQGMO *GetMsgOpts,
    CSIDD *DataOut,
    CSILONG *MsgLen,
    CSIQOS *QoS,
    CSILONG *ReturnCode,
    CSILONG *ReasonCode
);
```

Beware: As soon a message has been retrieved from a queue using the HL_mq_get() verb, this message is deleted from the queue.

Value of argument “Conn” is identical to the one obtained from HL_mq_conn().

Value of argument “Obj” is identical to the one obtained from HL_mq_open().

The “MsgDesc” argument contains, as input parameter, a set of attributes the message to retrieve must have and as output parameter the set of attributes the retrieved message actually has.

Method 1

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

When the application is in such a state that it is waiting for a report message, it has to set two attributes in the “MsgDesc” as follows:

```
MsgDesc.MsgId = CSIMQMI_NONE;
MsgDesc.CorrelId = {value of the MsgId in the original message, for
which a report is now awaited};
```

This initialisation is to be performed before each HL_mq_get() invocation.

Method 2

When the application is in such a state that it is waiting for any kind of message, it has to set two attributes in the “MsgDesc” as follows:

```
MsgDesc.MsgId = CSIMQMI_NONE;
MsgDesc.CorrelId = CSIMQCI_NONE;
```

This initialisation is to be performed before each HL_mq_get() invocation.

While the policy for reading messages from a queue with HL_mq_get() depends on the design of the application architecture, it is recommended though to use the Method 2 in combination with a priority value explained in attribute “Priority” of the “QoS”.

This means in practical terms that, within a set of messages read with a uniform “Priority” (see argument “QoS”), the messages will be read in their order of appearance.

They have then to be handled separately in line with the value of the “MsgDesc.MsgType” (either CSIMQMT_REQUEST, CSIMQMT_DATAGRAM or CSIMQMT_REPORT, as stated. Any message with its MsgType equal to CSIMQMT_REPORT must be matched to its own originator by the rule:

[report_message.CorrelId] is equal to [original_message.MsgId]

To be able to correlate a report with the related Information Exchange, it is recommended that the software controlling the sending CSI stack maintains a dynamic table that cross-references the state of a CSI message and its message identification. Message identification consists of:

- Value of field CSIMQMD.MsgId in the message sent by the sending NEA.
- Value of field CSIMQMD.CorrelId in a report message given by the sending NEA (exception, COA, expiration, COD).

The “GetMsgOpts” argument is a structure that controls the behaviour of the HL_mq_get() verb. The structure is shown in Table 45:

typedef struct tag		Initial value
CSIMQGMO{		
CSICHAR4	StrucId;	CSIMQGMO_STRUC_ID
CSILONG	Version;	CSIMQGMO_VERSION_1

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

CSILONG	Options;	
CSILONG	WaitInterval;	
CSILONG	Signal1;	(DNC)
CSILONG	Signal2;	(DNC)
CSICHAR48	DynamicQName;	(DNC)
} CSIMGMO;		

Table 45: CSIMQGM0 Object Descriptor

Notes

It is a design issue, related to the NDEA architecture, to choose between an applicative polling of a queue or a triggering mechanism initiated by CCN/CSI software, to be awakened upon a new message forthcoming in the queue. Regarding polling of a queue two processing mechanisms can be used:

- Constant CSIMQGM0_NO_WAIT is related to first choice;
- While CSIMQGM0_WAIT and value of WaitInterval set relate to second choice.

Whichever the choice taken, two precautions must be taken:

- “WaitInterval” cannot be set to CSIMQWI_UNLIMITED when “Options” has value CSIMQGM0_WAIT.
- When applicative polling is used (“Options” has value CSIMQGM0_NO_WAIT), then there must be a grace period foreseen in the application between two successive readings in the queue.

Value of argument “DataOut” represents the location of the data, when the value of the “MsgDesc.MsgType” is CSIMQMT_REQUEST or CSIMQMT_DATAGRAM. Otherwise (in the case of a CSIMQMT_REPORT) the CSIDD “DataOut” is left undefined (check this with the value of argument “MsgLen”, that must be 0L).

When a value for “DataOut” is defined, the attribute “Flags” of this CSIDD structure defines the way the information in CSIDD is to be represented.

Value of argument “MsgLen” represents the actual length in bytes of the application data in the retrieved message. It must be compared to the attribute “DataOut.DataLen”.

The argument “QoS” represents a CSIQOS structure that describes the particular handling that was applied on “DataOut”. Within this structure, only the “Priority” attribute is to be considered in order to satisfy to the **Rule** highlighted in X.I.2.2.6.

X.I.2.2.12.4 Browsing through a queue: HL_mq_browse()

```
CSILONG HL_mq_browse(
    CSIMQHCONN Conn,
    CSIMQHOBJS ObjDesc,
    CSIMQMD *MsgDesc,
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

CSIMQGMO    *GetMsgOpts ,
CSIDD       *DataOut ,
CSILONG     *MsgLen ,
CSIQOS      *QoS ,
CSILONG     *ReturnCode ,
CSILONG     *ReasonCode
);

```

All arguments used in this verb are explained as for the HL_mq_get() verb.

The HL_mq_browse() verb does not delete the message read from the queue. An explicit use of the verb HL_mq_delete() is required for deleting it.

Within the “QoS” structure, only the “Priority” attribute is to be considered in order to satisfy to the **Rule** highlighted in paragraph X.I.2.2.6.

X.I.2.2.12.5 Deleting an element from a queue: HL_mq_delete()

```

CSILONG HL_mq_delete(
CSIMQHCONN Conn ,
CSIMQHOBJ  ObjDesc ,
CSIMQMD    *MsgDesc ,
CSIMQGMO   *GetMsgOpts ,
CSIDD      *DataOut ,
CSILONG    *MsgLen ,
CSIQOS     *QoS ,
CSILONG    *ReturnCode ,
CSILONG    *ReasonCode
);

```

All arguments used in this verb are explained as for the HL_mq_browse() verb.

Important advice: a message may not be left indefinitely in a queue; it must be deleted at some time by using HL_mq_delete(). Otherwise, if the policy of the application is to neglect this message repeatedly, this message will always be in first position to be read after, hence will block the queue for reading any other message.

X.I.3 Environment

Messages on CCN are always exchanged between **Gateways**. In EMCS, two kinds of Gateways have to be considered:

- **National Gateways** used by all the MSAs to perform EMCS business.
- **Commission Gateways** used by Taxation and Customs Union DG or its contractors to operate the EMCS or to develop the CDEA.

MSAs and Taxation and Customs Union DG always have two Gateways: the **Operational Gateway** and the **Back-up Gateway**. On every Gateway, **Environments** are defined comprising of a set of queues that can be considered as a ‘layer’ in which messages are exchanged. The Operational Gateway contains the Operational Environment. The Back-up Gateway contains the Common Domain Testing Environment and the National Testing Environment. The Gateways and the respective environments are defined as follows:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- The Operational Gateway contains the:
 - The **Operational Environment** used to exchange the messages in EMCS operations in *Production Mode*.
- The **Back-up Gateway** contains the:
 - **Common Domain Testing Environment** used to exchange messages for testing purposes on the Common Domain. This environment is used for National Compliance Testing (*Mode-1*) and Conformance Testing (*Mode-2*) with TA. This environment is also used for International Testing (*Mode-3* and *Mode-3+*) between two or more countries.
 - **National Testing Environment**, which is an optional environment, used to exchange messages in loopback mode so that messages are not exchanged on the Common Domain with other Gateways. This environment can be used for Integration Testing (*Mode-0*) with a National Excise Test Application¹⁰ (NETA) deployed at the premises of MSAs.

In every environment, **queues** have to be defined on the different Gateways. It has to be noted that messages should only be exchanged between queues belonging to the same Environment.

Every EMCS queue always has the following syntax:

<QUEUE NAME.XXXX@GATEWAY NAME>

Where “XXXX” is either “SEED” for SEED queues and “EMCS” for all other queues.

The queues can be divided in several groups based upon their function:

- **‘Core flow’** is used for the messages marked in Table 46.
- **‘Administration’** is used for the IE906 and IE917 refusal messages resulting from erroneous messages received from the ‘Core flow’ queue. It is also used for the Information Exchanges between NEAs and CS/MISE, namely the IE770 and IE917 messages.
- **‘SEED’** is used for IE701, IE702, IE713, IE714, IE734, IE742, and IE917 messages exchanged with SEED. It should be noted that in case an NEA sends a syntactically invalid message to SEED, the NEA will receive the IE917 refusal message from the ‘SEED’ queue (i.e. the IE917 will not be received from the ‘Administration’ queue as it is the case for the syntactically invalid message exchanged via the ‘Core flow’ queue).

¹⁰ NETA is a nationally developed application and it should not be confused with TA, which is a centrally developed application. TA is deployed only centrally and it cannot be deployed at the premises of MSAs.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- **‘Reports’** is used for the CCN/CSI Reports (IE908, IE909, IE910 and IE911). The CCN/CSI reports are returned for every message type. It is received by a sending application in the queue that was indicated by the QoS "ReplyToQ" argument when sending the Information Exchange. It is highly recommended to request these reports in the associated Reports queue. In this manner, pending reports can be recognised quickly and problems can be identified in a straightforward manner. It is recommended to use the Reports queue solely for storing those CCN/CSI reports.
- **‘Technical Statistics’** is used for CCN/CSI technical statistics from CCN/TC.¹¹
- **‘Audit’** is used for CCN/CSI audit files from CCN/TC.¹²

IE	IE Name	Reference
701	Common Request	C_REQ_SUB
702	Refusal of Common Request	C_REQ_REF
717	Control Report	C_CCR_DAT
721	Administrative Cooperation Common Request	C_COO_SUB
801	E-AD	C_EAD_VAL
802	Reminder Message for Excise Movement	C_EXC_REM
803	Notification of Diverted e-AD	C_EAD_NOT
807	Interruption of Movement	C_STP_NOT
810	Cancellation of e-AD	C_CAN_DAT
813	Change of Destination	C_UPD_DAT
818	Accepted or Rejected Report of Receipt/Export	C_DEL_DAT
819	Alert or Rejection of an e-AD	C_REJ_DAT
820	History Results	C_HIM_RES
821	List of e-AD as Result of a General Query	C_LST_VAL
829	Notification of Accepted Export	C_EXP_NOT
837	Explanation on Delay For Delivery	C_DEL_EXP
839	Customs Rejection of e-AD	C_CUS_REJ
840	Event Report	C_EVT_DAT
861	Basis for Recovery of Duties	C_RES_DAT
867	Administrative Cooperation Results	C_COO_RES
868	Answer Message	C_COO_ANS
869	Reminder Message for Administrative Cooperation	C_COO_REM
871	Explanation on Reason For Shortage	C_SHR_EXP
904	Status Request	C_STD_REQ
905	Status Response	C_STD_RSP
934	Data Packaging	C_PAC_DAT

Table 46: Information Exchanges exchanged via the ‘Core flow’ queue

¹¹ Please note that no queues are defined for MSAs for this function.

¹² Please note that no queues are defined for MSAs for this function.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

In the following chapters, the queues per environment are defined as well as the actual names to be used for the queues and the Gateways. Chapter X.I.3.1 defines the national queues and Gateways and Chapter X.I.3.2 defines the Taxation and Customs Union DG queues and Gateways.

X.I.3.1 National Gateways

X.I.3.1.1 Queue Name

Every National Gateway has to be configured to contain the following queues as shown in Table 47:

Mode	Queue Function	Queue Name
Production	Core flow	CORE-XML-QUE.EMCS
	Administration	ADMIN-XML-QUE.EMCS
	Reports	REPORT-QUE.EMCS
	SEED	NA-XML-QUE.SEED
		NA-REPORT-QUE.SEED
	Technical Statistics	-
Audit	-	
Mode-3 and Mode-3+	Core flow	CORE-XML-RIT-QUE.EMCS
	Administration	ADMIN-XML-RIT-QUE.EMCS
	Reports	REPORT-RIT-QUE.EMCS
	SEED	-
	Technical Statistics	-
	Audit	-
Mode-2	Core flow	TA-NEA-CORE-XML-RCT-QUE.EMCS
	Administration	TA-NEA-ADMIN-XML-RCT-QUE.EMCS
	Reports	TA-NEA-REPORT-RCT-QUE.EMCS
	SEED	NA-XML-RCT-QUE.SEED
		NA-REPORT-RCT-QUE.SEED
	Technical Statistics	-
Audit	-	
Mode-1	Core flow	TA-NEA-ED-CORE-XML-RCT-QUE.EMCS
	Administration	TA-NEA-ED-ADMIN-XML-RCT-QUE.EMCS
	Reports	TA-NEA-ED-REPORT-RCT-QUE.EMCS
	SEED	NA-XML-RCT-QUE.SEED
		NA-REPORT-RCT-QUE.SEED
	Technical Statistics	-
Audit	-	
Mode-0	Core flow	CORE-XML-LCT-QUE.EMCS
		CORE-xx-LCT-QUE.EMCS
	Administration	ADMIN-XML-LCT-QUE.EMCS
		ADMIN-xx-LCT-QUE.EMCS
	Reports	REPORT-LCT-QUE.EMCS
		REPORT-xx-LCT-QUE.EMCS

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

	SEED	NA-XML-LCT-QUE.SEED
		NA-XML-xx-LCT-QUE.SEED
		NA-REPORT-LCT-QUE.SEED
		NA-REPORT-xx-LCT-QUE.SEED
	Technical Statistics	-
	Audit	-

Table 47: Queue Names for National Gateways

Notes:

- In the above table, where “xx” can take the values 01 - 10 to indicate 10 different queues for use by a National Excise Test Application in National Testing environment.
- The queue names are applicable for EMCS.
- The report queues for Mode-1 and Mode-0 are optional.

X.I.3.1.2 Gateway Site Names

The National Gateway Names (valid at the time of publication) are defined in Table 48:

Country	Gateway Name
AT	CUST.AT
BE	CUST.BE
BG	CUSTTAX.BG
CY	CUSTTAX.CY
CZ	CUST.CZ
DE	CUST.DE
DK	CUSTTAX.DK
EC	CCN.TC
EC	DGXXI.EC
EE	CUSTTAX.EE
EL	CUSTTAX.EL
ES	CUSTTAX.ES
FI	CUST.FI
FR	CUSTTAX.FR
GB	CUSTTAX.GB
HU	CUSTTAX.HU
IE	CUSTTAX.IE
IT	CUST.IT
LT	CUSTTAX.LT
LU	CUSTTAX.LU
LV	CUSTTAX.LV
MT	CUSTTAX.MT
NL	CUSTTAX.NL
PL	CUSTTAX.PL
PT	CUSTTAX.PT
RO	CUSTTAX.RO

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Country	Gateway Name
SE	TAX.SE
SI	CUSTTAX.SI
SK	CUSTTAX.SK
HR	CUSTTAX.HR

Table 48: National Gateway names

X.I.3.2 Taxation and Customs Union DG Gateways (DG TAXUD)

X.I.3.2.1 Queue Name

The names of the Queues are defined as shown in Table 49:

Mode	Queue Function	Queue Name	
Production	Core flow	-	
	Administration	ADMIN-XML-QUE.EMCS	
	Reports	REPORT-QUE.EMCS	
	SEED		CS-XML-QUE.SEED
			CS-REPORT-QUE.SEED
	Technical Statistics	STAT-QUE.EMCS	
	Audit	AUDIT-QUE.EMCS	
Mode-3 and Mode-3+	-		
Mode-2	Core flow	TA-CORE-role-XML-xx-RCT-QUE.EMCS	
	Administration	TA-ADMIN-role-XML-xx-RCT-QUE.EMCS	
	Reports	TA-REPORT-xx-RCT-QUE.EMCS	
	SEED		CS-XML-RCT-QUE.SEED
			CS-REPORT-RCT-QUE.SEED
	Technical Statistics	STAT-RCT-QUE.EMCS	
	Audit	AUDIT-RCT-QUE.EMCS	
Mode-1	Core flow	TA-ED-CORE-role-XML-xx-RCT-QUE.EMCS	
	Administration	TA-ED-ADMIN-role-XML-xx-RCT-QUE.EMCS	
	Reports	TA-ED-REPORT-xx-RCT-QUE.EMCS	
	SEED		CS-XML-RCT-QUE.SEED
			CS-REPORT-RCT-QUE.SEED
	Technical Statistics	STAT-RCT-QUE.EMCS	
	Audit	AUDIT-RCT-QUE.EMCS	
Mode-0	-		

Table 49: Queue Names for Taxation and Customs Union DG Gateways

Notes:

In the above table:

- “role” is the roles for use by TA in Common Domain Testing environment defined in the TA SRD [R12]; and

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- “xx” is a value for use by TA in Common Domain Testing environment defined in Table 50.

Country	xx
CY	01
DE	02
EE	03
ES	04
IT	05
MT	06
NL	07
BG	08
CZ	09
AT	10
BE	11
SK	12
DK	13
FI	14
FR	15
GB	16
EL	17
HU	18
IE	19
LT	20
LU	21
LV	22
PL	23
PT	24
RO	25
SE	26
SI	27
HR	28

Table 50: National Gateway name

X.I.3.2.2 Gateway Names

The Gateway name of the Taxation and Customs Union DG Gateways is always ‘DGXXI.EC’.

X.I.3.3 Queue usage Overview

The following chapters provide an overview of the usage of the queues in the different environments by the EMCS.

In every figure, the left side is the EMCS and the right side shows the application with which it is communicating. To be noted:

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

If the CCN/CSI network is coloured grey, it indicates the Back-up Gateway has to be used. Otherwise, the Operational Gateway has to be used.

The queues that are coloured grey, are those defined at the Taxation and Customs Union DG Gateway. The others are defined at the NDEA Gateways.

X.I.3.3.1 Operational Environment

The following diagrams graphically depict the normal operations of a NEA that interacts with another NEA, with SEED, or with CS/MISE.

The same sets of operations are foreseen for NEAs.

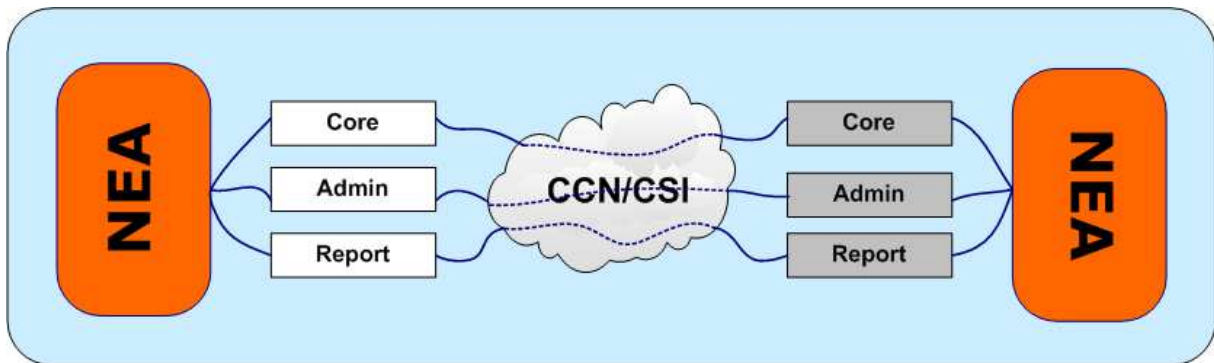


Figure 230: Normal Operations with a NEA

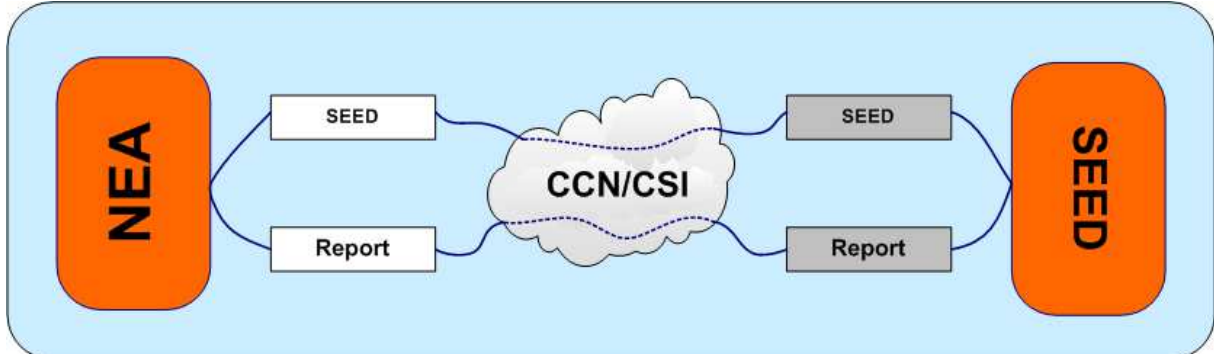


Figure 231: Normal Operations with SEED

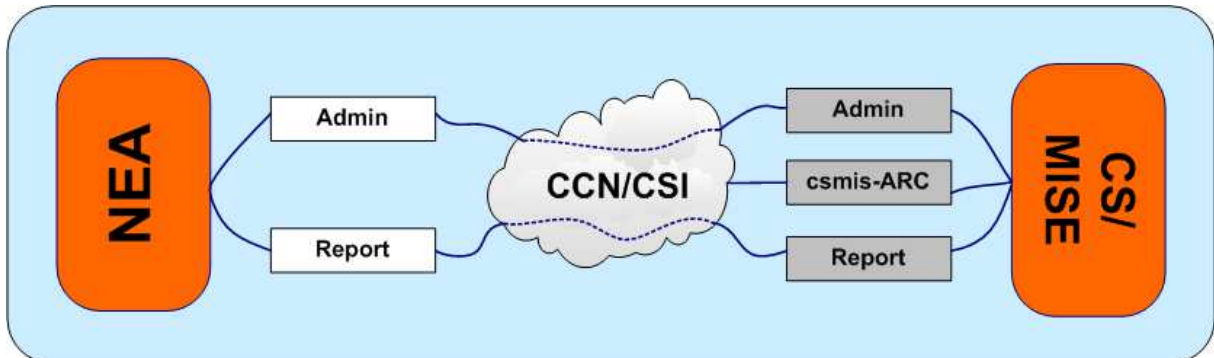


Figure 232: Normal Operations with CS/MISE

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.3.3.2 Common Domain Testing Environment

The following diagrams graphically depict the operations for Conformance Testing of a NEA.

The same sets of operations are foreseen for NEAs.

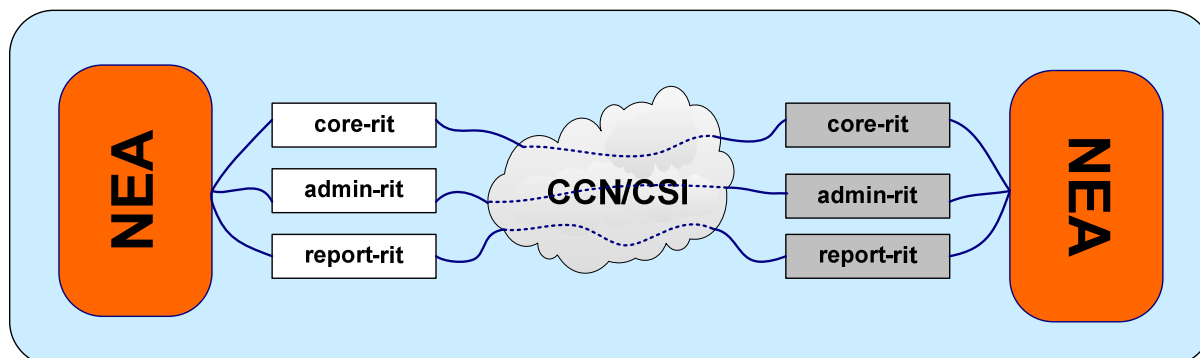


Figure 233: International Testing with another NEA

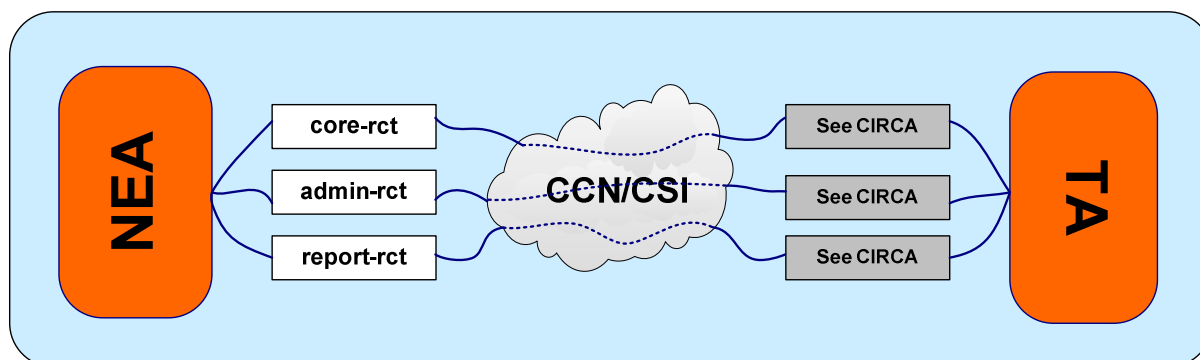


Figure 234: Conformance Testing

X.I.4 Recommended Use of CCN/CSI

This section illustrates the use of CCN/CSI in a send and receives routine. This example is only a guide for implementation. It is up to each MSA to implement its routines for sending and receiving. The specifications of both routines show the general sequence of synchronous interaction between the application and the corresponding CSI component on the Gateway (“Remote API Proxy”).

The routines are specified in a C-like pseudo-code. The setting of a number of parameters of CSI verbs is not shown when these are not relevant for this explanatory purpose.

X.I.4.1 Main routines

Typical execution phases are as follows:

6. Program Connection phase:
 - Program binding to the CSI stack (HL_bind).
 - Establishment of a security context (HL_init_sec_context).
 - Connection to the local Queue Manager (HL_mq_conn).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

7. Sending phase:

- Opening of a message queue with the appropriate options (HL_mq_open).
- Encode the data descriptor passed by the application into another data descriptor, which will be handled by the Sending function, by using a Presentation profile which corresponds to the Information Exchange type (HL_encode).
- Send a message to a remote queue that has been opened for output (HL_mq_put). If a queue has not yet been opened, the verb HL_mq_put1 can be used. This latter verb also implies closing the queue after inserting the message into it.
- Closing the opened message queue (HL_mq_close).

8. Receiving phase:

- Opening of a message queue with the appropriate options (HL_mq_open).
- Destructive extraction of a message from a local queue (HL_mq_browse, HL_decode, and HL_mq_delete). This sequence of verbs is recommended, rather than the verb HL_mq_get, to overcome deletion of a message if it does not fit in a memory buffer.
- Decode the received data descriptor into another data descriptor, for application usage. No application profile is to be communicated to this verb. This verb checks the correctness of the CodePage and HostFormat used.
- Closing the opened message queue (HL_mq_close).

9. Program Disconnection phase:

- Disconnection from the Queue Manager (HL_mq_disc).
- Destruction of the security context (HL_delete_sec_context).
- Disconnection from the CSI stack (HL_unbind).

These four execution phases can be executed in a sequence as is for instance shown by next figure. Other sequences are also possible, e.g. parallel sending and reception of messages or parallel sending of messages to different queues. Another option is to combine the Sending and Receiving phases into one routine. This routine may first send and secondly read messages out of queues.

It is recommended to distinguish between priorities in sending and receiving, e.g. first browsing a queue for high priority messages before the messages with normal priority are received from the same queue.

For reception, all available queues for reading identified via the Access Control List need to be browsed. A sequence of browsing and reading is up to each application developer.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

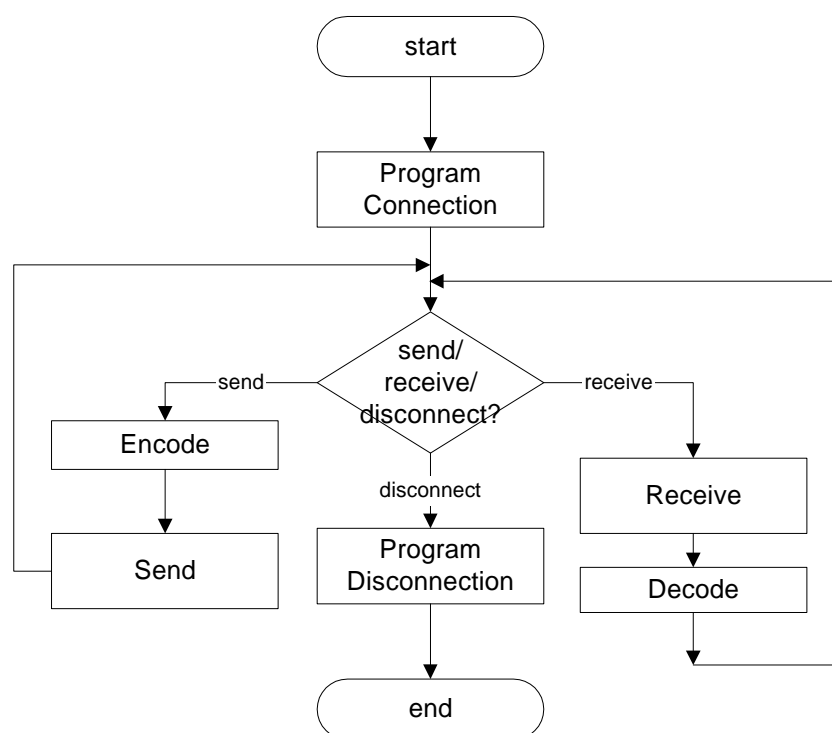


Figure 235: A possible sequence for using CSI verbs

Message queue opening and closing and queue access verbs, as shown in the Sending and Receiving phases, can be inter-mixed in any sequence with the restriction that queues must be properly opened before messages can be written to or read from it.

All program interactions are with the local Queue Manager running in the local CCN gateway. As a consequence, the transmission and processing delays do not slow sending or receiving programs and the response times are significantly improved.

Errors related to the use of CCN/CSI are specified in the appropriate CCN/CSI documentation. They have to be handled by an application program connecting to the CSI stack.

The next examples use the arguments (CSILONG) **pReturnCode**, **pReasonCode** as output parameters of a CSI verb. The first indicates success or failure and, in case of failure, the second gives the reason for failure. Working code should check these values after every CCN/CSI call.

X.I.4.2 Program connection

The following pseudo-code is given for connection of an application named “NEA” to the CCN Gateway.

Connect (out MQHCONN, Returncode)

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- The result of the connect is an identifier that needs to be used for sending, receiving, and disconnecting a session with the CCN Gateway (MQHCONN). The result of the execution of this routine is given in Returncode. Errors for executing verbs can possibly be hidden in the connect routine by retrying to connect to the Gateway more than once. Some errors, e.g. there is no validated user password, which require a systems administrator to take action, need to be passed to the application.

HL_bind (in “NEA”, “”, out CSIQOS, pReturnCode, pReasonCode)

- “NEA” is now bound to the CSI stack. Default ProxyName was used (the empty string “”).
- Default QoS for “NEA” is returned by the Gateway in parameter CSIQOS.
- The first CSILONG indicates the success or failure of the verb execution, whereas the second specifies an error reason. If CSILONG indicates a failure, a retry of Connect may be tried, depending on the error reason.
- The application is in charge of obtaining security credentials: UserName equals “NEA” and, “UserPassword” and “ApplicationKey” have to be valid parameter values checked by the CCN Gateway. The variable SecurityInfo, with type CSISECINFO, has been filled as explained above in paragraph X.I.2.2.9.

HL_init_sec_context (in SecurityInfo, out pReturnCode, pReasonCode)

- Security context between Application Platform and CCN gateway is established. If the security context could not be established, the CSILONG parameters contain the success or failure with the associated reason. The application key and username is checked by the X.500 directory of the Gateway. Both parameters need to be known to the application.

HL_mq_conn (in Mqman, out MQHCONN, pReturnCode, pReasonCode)

- Application is connected to the message queue management system identified by Mqman. The Gateway returns the status of the execution of the verb and an identifier to be used during exchange of information between the “NEA” and the Gateway (MQHCONN).

X.I.5 Configuration Information

X.I.5.1 Introduction

This chapter presents the steps that need to be performed in order to set up the parameters on the CCN Gateways.

This chapter is divided into:

- The information to be prepared by a MSA for all objects that are specific to each MSA.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- The information to be prepared by EMCS-CO and that are required to ensure end-to-end interoperability.

Chapter 5 of document [R8] presents a Responsibility Model that distributes the choices to be taken between several roles. The roles applicable for EMCS are:

- **CDIA:** the CCN/TC Directory Administrator, responsible for the central management of the CCN directory.
- **CASO:** the Central Application Security Officer, responsible for security issues that concern a given application of the Taxation and Customs Union DG, running over the CCN/CSI system. This is actually the EMCS-CO.
- **CSO:** CCN-TC Central Security Officer.
- **LAD:** a Local Application Designer, responsible for the design of a NDEA program. In the case of a CDEA, the contractor designer has to further subdivide the design issues between what the CDEA was able to decide and what is left to the MSA development to decide.
- **LSO:** the Local Security Officer, responsible for security issues for a NDEA.
- **LSYA:** the Local System Administrator for the NDEA infrastructure, responsible for the system management.
- **LAA:** the Local Application Administrator for the NDEA infrastructure, responsible for the management of the CCN directory data related to the local users of NDEA: this amounts to maintaining the list of UserIds with respect to UserProfiles.

X.I.5.2 Configuration information to be provided by the MSA

Security aspects for EMCS are defined in [R2] and [A10].

Concerning CCN/CSI exchanges, a number of standard security features of the CCN/CSI network will be used. The MSA only needs to set up the proper security configuration on their gateways, in co-ordination with the CCN/TC. For CCN/CSI exchanges, document [A7] is containing additional information on security aspects.

X.I.5.2.1 Collection of External Configuration Data

The words ‘Entity’ and ‘Attributes’ used in Table 51 refer to the ERD defined in [R8].

The associated values depend on the technical infrastructure that exists within a MSA.

These values have to respect the “Formatting Rules” defined in heading 5.3 of [R8].

The values have to be entered onto forms in Annex C of [R8].

It is necessary to cooperate with EMCS-CO for the practical handling of these forms as shown in Table 51.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Entity	Attribute	Provided by	Managed by
Application	ccnApplicationName	LAD	CDIA
	ccnApplicationType	LAD	CDIA
	ccnAddress	LAD	CDIA
	ccnAddressType	LAD	CDIA
	ccnAuthorizedSecurityMechanisms	LAD	CDIA
	ccnDefaultSecurityMechanisms	LAD	CDIA
	ccnDataRepresentationRules	LAD	CDIA
	ccnDefaultCodePage	LAD	CDIA
	ccnApplicationActivationMode	LAD	CDIA
	ccnApplicationExchangeMode	LAD	CDIA
	ccnConversationalModeEnabled	LAD	CDIA
	ccnApplicationSecurityKey	LSO	LSA or LAA
	ccnDefaultQOS	LAD	CDIA
	ccnOperationalStatus	LSYA	CDIA
	ccnPlatformName	LAD	CDIA
ccnRAPName	LAD	CDIA	
Platform	ccnPlatformName	LAD	CDIA
	ccnAddress	LAD	CDIA
	ccnAddressType	LAD	CDIA
	ccnAuthorizedSecurityMechanisms	LAD	CDIA
	ccnDefaultSecurityMechanisms	LAD	CDIA
	ccnDataRepresentationRules	LAD	CDIA
	ccnDefaultCodePage	LAD	CDIA
	ccnOperationalStatus	LSYA	CDIA
Queue	ccnQueueName	LAD	CDIA
	ccnTriggerEnabled	LAD	CDIA
	ccnTriggerType	LAD	CDIA
	ccnMaxQueueDepth	LAD	CDIA
	list of ccnApplicationName	LAD	CDIA
Remote API Proxy	ccnRAPName	LAD	CDIA
	ccnMinNbOfRAPInstances	LAD	CDIA
	ccnMaxNbOfRAPInstances	LAD	CDIA
User	ccnUserName	LSO	LSA or LAA
	ccnUserPassword	LSO	LSA or LAA
	ccnUserDisabled	LSO	LSA or LAA
	ccnOrganisationName	LSO	LSA or LAA
	list of ccnUserProfileId	LSO	LSA or LAA

Table 51: External Configuration Data defined by MSA

X.I.5.2.2 Message configuration procedure

The general message configuration procedure is performed by EMCS-CO.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

Note that the information ‘ccnDataRepresentationRules’ and ‘ccnDefaultCodePage’ presented in paragraph X.I.5.2.1 will be used by the CCN Technical Centre to generate files specific to the NDEA development.

X.I.5.3 Configuration information to be provided by the EMCS-CO

X.I.5.3.1 Collection of External Configuration Data

Entity	Attribute	Provided by	Managed by
Application	ccnDefaultQOS	LAD-CAD	CDIA
CCN/CSI organisation	ccnOrganisationName	CDIA	CDIA
CCN gateway	ccnGatewayName	CDIA	CDIA
Message	ccnMessageId	CAD	CDIA
	ccnMessageFormalDefinition	CAD	CDIA
Queue	list of ccnUserProfileId	CASO	CSA
	list of ccnUserProfileId	CASO	CSA
	ccnGatewayName	CAD	CDIA
Remote API Proxy	ccnGatewayName	CAD	CDIA
User profile	ccnUserProfileId	CASO	CSA

Table 52: External Configuration Data defined by EMCS-CO

The attributes values to be configured on EMCS-wide scale by EMCS-CO are detailed below.

X.I.5.3.1.1 ccnDefaultQOS

As explained in the value configured on all Gateways for all applications may be overridden upon each call (HL_mq_put() or HL_mq_put1()) performed by an application. The values from Table 53 will be entered into CCN/CSI application configuration form-part 4:

Configuration of default QoS	
Priority:	5
ReportRequest:	Exception Expiration Confirm on Arrival Confirm on Delivery PassMessageId PassCorrelId
ReplyToQ:	left empty
Integrity:	Forbidden
Confidentiality:	Forbidden
Compression:	Forbidden
CompressionId:	LZW
DegradedMode:	NotAllowed -- N/A anyway
CoT:	DEFAULTCOT

Table 53: Configuration of default QoS

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

X.I.5.3.1.2 ccnGatewayName

See Table 52.

X.I.5.3.1.3 ccnOrganisationName

See Table 51 and Table 52.

X.I.5.3.1.4 ccnMessageId

- “CD701A-MSG.emcs”
- “CD702A-MSG.emcs”
- “CD713A-MSG.emcs”
- “CD714A-MSG.emcs”
- “CD721A-MSG.emcs”
- “CD734A-MSG.emcs”
- “CD742A-MSG.emcs”
- “CD770A-MSG.emcs”
- “CD801A-MSG.emcs”
- “CD802A-MSG.emcs”
- “CD803A-MSG.emcs”
- “CD807A-MSG.emcs”
- “CD810A-MSG.emcs”
- “CD813A-MSG.emcs”
- “CD818A-MSG.emcs”
- “CD819A-MSG.emcs”
- “CD820A-MSG.emcs”
- “CD821A-MSG.emcs”
- “CD829A-MSG.emcs”
- “CD837A-MSG.emcs”
- “CD839A-MSG.emcs”

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

- “CD840A-MSG.emcs”
- “CD861A-MSG.emcs”
- “CD867A-MSG.emcs”
- “CD868A-MSG.emcs”
- “CD869A-MSG.emcs”
- “CD871A-MSG.emcs”
- “CD904A-MSG.emcs”
- “CD905A-MSG.emcs”
- “CD934A-MSG.emcs”
- “CD906A-MSG.emcs”
- “CD917A-MSG.emcs”

X.I.5.3.1.5 ccnMessageFormalDefinition

See paragraph X.I.5.3.2.

X.I.5.3.1.6 ccnUserProfileId

Syntax for defining the ccnUserProfileId is:

[UserProfileId][ApplicationModeSuffix]-PRF.XXXX

Where “XXXX” is either “SEED” for SEED profiles and “EMCS” for all other profiles.

The names of the UserProfileId per queue are defined as shown in Table 54 (for more information regarding queues please refer to Section X.I.3 Environment). For detailed implementation of the UserProfileIds for SEED and TA, please refer to the corresponding CTODs (Conformance Test Organisation Documents).

Queue Name	CCN profile	Access
CORE-XML-QUE.EMCS	CORE-XML-PRF.EMCS	R/W
ADMIN-XML-QUE.EMCS	ADMIN-XML-PRF.EMCS	R/W
REPORT-QUE.EMCS	REPORT-PRF.EMCS	R/W
NA-XML-QUE.SEED	NDEA-XML-PRF.SEED	R
CS-XML-QUE.SEED	NDEA-XML-PRF.SEED	W
NA-REPORT-QUE.SEED	NA-REPORT-PRF.SEED	R
CS-REPORT-QUE.SEED	NA-REPORT-PRF.SEED	W
CORE-XML-RIT-QUE.EMCS	CORE-XML-RIT-PRF.EMCS	R/W

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

ADMIN-XML-RIT-QUE.EMCS	ADMIN-XML-RIT-PRF.EMCS	R/W
REPORT-RIT-QUE.EMCS	REPORT-RIT-PRF.EMCS	R/W
TA-NEA-CORE-XML-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-CORE-role-XML-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
TA-NEA-ADMIN-XML-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-ADMIN-role-XML-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
TA-NEA-REPORT-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-REPORT-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
NA-XML-RCT-QUE.SEED	NDEA-RCT-PRF.SEED	R
CS-XML-RCT-QUE.SEED	NDEA-XML-RCT-PRF.SEED	W
NA-REPORT-RCT-QUE.SEED	NDEA-RCT-PRF.SEED	R
CS-REPORT-RCT-QUE.SEED	NA-REPORT-RCT-PRF.SEED	W
TA-NEA-ED-CORE-XML-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-ED-CORE-role-XML-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
TA-NEA-ED-ADMIN-XML-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-ED-ADMIN-role-XML-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
TA-NEA-ED-REPORT-RCT-QUE.EMCS	TA-READ-RCT-PRF.EMCS	R
TA-ED-REPORT-xx-RCT-QUE.EMCS	TA-WRITE-RCT-PRF.EMCS	W
CORE-XML-LCT-QUE.EMCS	CORE-XML-LCT-PRF.EMCS	R
CORE-xx-LCT-QUE.EMCS	CORE-xx-LCT-PRF.EMCS	W
ADMIN-XML-LCT-QUE.EMCS	ADMIN-XML-LCT-PRF.EMCS	R
ADMIN-xx-LCT-QUE.EMCS	ADMIN-xx-LCT-PRF.EMCS	W
REPORT-LCT-QUE.EMCS	REPORT-LCT-PRF.EMCS	R
REPORT-xx-LCT-QUE.EMCS	REPORT-xx-LCT-PRF.EMCS	W
NA-XML-LCT-QUE.SEED	NA-XML-LCT-PRF.SEED	R
NA-XML-xx-LCT-QUE.SEED	NA-XML-xx-LCT-PRF.SEED	W
CS-XML-LCT-QUE.SEED	NDEA-XML-LCT-PRF.SEED	W
NA-REPORT-LCT-QUE.SEED	NA-REPORT-LCT-PRF.SEED	R
NA-REPORT-xx-LCT-QUE.SEED	NA-REPORT-xx-LCT-PRF.SEED	W
CS-REPORT-LCT-QUE.SEED	NA-REPORT-LCT-PRF.SEED	W

Table 54: List of user profiles

In the above table “xx” can take the value 01 - 10 to indicate 10 different queues.

X.I.5.3.2 Message configuration procedure for EMCS

Table 55 is the IDL definition for all EMCS messages exchanged over the Common Domain that have a relevant message body: the list of these messages is in Table 41. The report messages have not to be defined with an IDL description.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

The EMCS-CO must have this IFL definition compiled by the CCN-TC service and must forward to each MSA the files obtained from this compilation. Section 6 of document [R8] has to be followed for this process.

```

/* CCNIDL definition of EMCS messages for Phase 3 */
[ version (1.0) ]
interface CD701A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD702A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD713A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD714A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD721A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD734A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD742A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD770A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD801A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD802A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD803A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD807A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD810A-MSG.emcs
{
typedef byte RawBuffer ;
}

```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

interface CD813A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD818A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD819A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD820A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD821A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD829A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD837A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD839A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD840A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD861A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD867A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD868A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD869A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD871A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD904A-MSG.emcs
{
typedef byte RawBuffer ;
}

```


EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section X - Transport of Messages via CCN/CSI	

```

}
interface CD905A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD934A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD906A-MSG.emcs
{
typedef byte RawBuffer ;
}
interface CD917A-MSG.emcs
{
typedef byte RawBuffer ;
}

```

Table 55: IDL definition of CCN messages for EMCS

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Section XI TRANSPORT OF MESSAGES VIA SOAP/HTTP

This chapter defines the transport of messages via SOAP/HTTP that can be used by a MSA to invoke the Common Domain Central Services Web Services. It is decomposed in seven sub-sections:

The first section discusses the network topology that enables MSA client and the Common Domain Central Services interaction over SOAP/HTTP.

The second section details the CCN configuration and related responsibility model and procedures.

The third section provides a brief introduction to the standards that are applicable to web service invocation over SOAP/HTTP: SOAP, HTTP, XML-RPC, WSDL, WS-Security and UDDI. The fourth section ‘Recommended Usage’ provides an explanation of how a MSA should use Common Domain Central Services Web Services:

- The “SOAP Message Structure” to create;
- The “SOAP Message Exchange Patterns” to be used;
- How a MSA must invoke web services of Central Services over SOAP/HTTP.

The last two sections outline two important features of web services that are relevant to the SOAP/HTTP transport: “Web Service Transactions” and “Web Service Security”.

Sub-Section XI.I Topology

XI.I.1 HTTP Transport

The topology of the HTTP transport over CCN can be represented as follows:

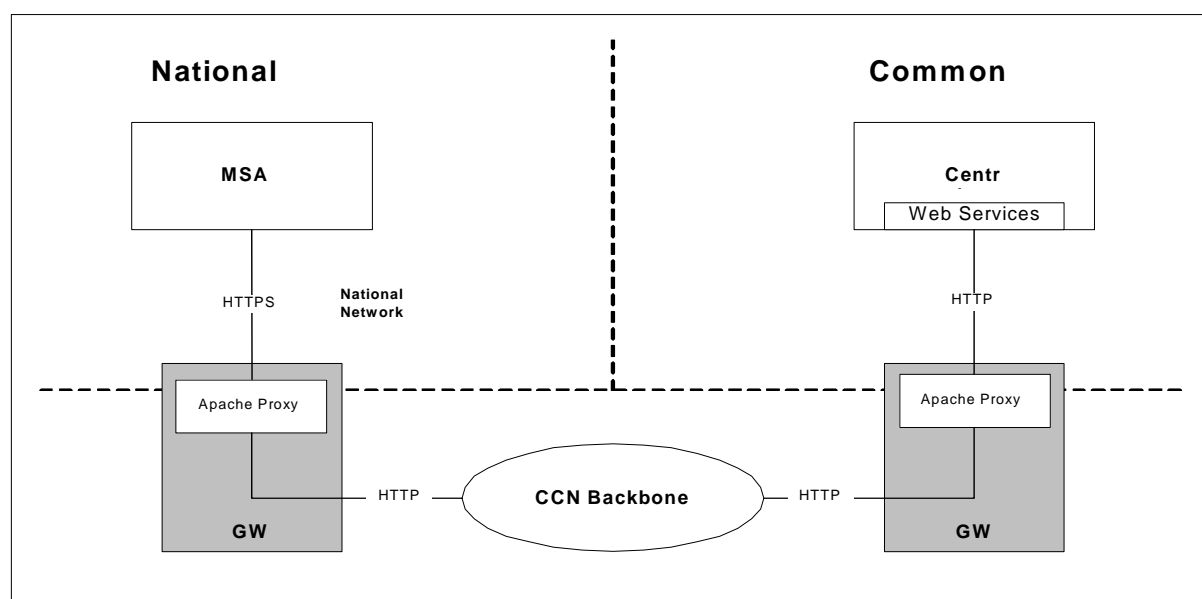


Figure 236: HTTP over CCN Transport Topology

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

SOAP is the message format used in the invocation of Central Services Web Services over a HTTP transport. SOAP messages created by MSAs are sent via HTTP over CCN through the Gateways Apache Proxies, to Central Services Web Services.

XI.I.2 Uniform Resource Identifier

Common Domain Central Services Web Services are located by *URIs*. Following the generic syntax or URIs, the derived syntax for each Common Domain Central Services web service is defined as follows:

$$\langle \text{scheme} \rangle : // \langle \text{server} \rangle : \langle \text{port} \rangle / \langle \text{path} \rangle$$

Where:

- $\langle \text{scheme} \rangle$ is always “http” or “https” - only SOAP/HTTP is supported by Central Services;
- $\langle \text{server} \rangle$ is an environment specific logical name that identifies a HTTP server. See sub-section “Environments” below for more details;
- $\langle \text{port} \rangle$ represents the TCP/IP port that the HTTP transport runs over. As with the $\langle \text{server} \rangle$ element, the HTTP port varies between “Environments”;
- $\langle \text{path} \rangle$ represents a relative path to the Central Services web service resource.

Section XI.I.3 below discusses the environments that support for the SOAP/HTTP transport, including the environment specific values for $\langle \text{server} \rangle$ and $\langle \text{port} \rangle$.

It also lists the relative URIs for each Central Services web service and specifies the values for the $\langle \text{path} \rangle$ element - “ $\langle \text{WebApp} \rangle / \langle \text{RelativePath} \rangle / \langle \text{WebService} \rangle . \text{ws}$ ” - that does not change between environments.

XI.I.3 Environments - Web Service Relative Path

The Web Service relative path - that is, the value for $\langle \text{path} \rangle$ in the full URI syntax is defined by the following syntax:

$$\langle \text{path} \rangle ::= \langle \text{WebApp} \rangle / \langle \text{RelativePath} \rangle / \langle \text{WebService} \rangle$$

Where:

- $\langle \text{WebApp} \rangle$ is the name of the Central Services Web Services web application;
- $\langle \text{RelativePath} \rangle$ is a logical path relative to $\langle \text{WebApp} \rangle$;
- $\langle \text{WebService} \rangle$ is the name of the Central Services web service.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Web Service relative path do not change between environments.

Process Area	Web Services	Relative URI
Reference Data	Re-synchronisation of Reference Data	/CentralServicesWS/SEED/RetrieveOrExtractEntity.jws
SEED data	Re-synchronisation of SEED data	/CentralServicesWS/SEED/RetrieveOrExtractEntity.jws
	Dissemination of SEED data (Reception of Updates from the MSAs)	/CentralServicesWS/SEED/MaintainEntity.jws

Table 56: Central Services Web Service Relative URIs by Processing Area

Here are some examples of Central Services web service URIs:

- <http://cs.dgtaxud.ec/CentralServicesWS/SEED/RetrieveOrExtractEntity.jws>

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Sub-Section XI.II CCN Configuration

The CCN configuration data to provide are the list of user profiles and the URIs accessible by these profiles through CCN/HTTP. This configuration is required both for Web Services access and for access to the Web interface of the application.

XI.II.1 Responsibility Model

The following roles are defined for the provisioning of the CCN/HTTP attribute value(s):

- Central Application Designer (CAD), responsible for the design of a given DG TAXUD application running over the CCN/CSI system;
- Local Security Officer (LSO), responsible for security issues within a given CCN/CSI organisation.

The following roles are defined for the management of the attribute value(s), i.e. for the registration/modification of the attribute value(s) in the CCN directory:

- CCN-TC Directory Administrator (CDIA), responsible for the central management of the CCN directory.

XI.II.2 CCN/HTTP Configuration Data

For each environment, there are five profiles defined for MSAs¹³:

- ELO: An ELO has access to the maintenance of the Economic Operators of its MS;
- ADM: Administrators maintain the notification profiles of the users of their MSA;
- READ: Read-only users may only read the information;
- WS-ELO: This profile gives access to the Web Services for maintenance of Economic Operators;
- WS-READ: This profile gives read-only access to the Web Services.

The following profiles have to be defined on National Gateways to control the access to SEED Web Services:

ccnUserProfileId
WS-ELO-PRF.SEED
WS-READ-PRF.SEED
WS-ELO-RCT-PRF.SEED
WS-READ-RCT-PRF.SEED
WS-ELO-RST-PRF.SEED

¹³ As Commission's users will not access SEED through CCN, their profiles are not defined here.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

ccnUserProfileId
WS-READ-RST-PRF.SEED

Table 57: User Profiles for National Gateways access to Central Services Web Services

The following profiles have to be defined on National Gateways to control the access to the SEED Web interface:

ccnUserProfileId
ELO-PRF.SEED
READ-PRF.SEED
ADM-PRF.SEED
ELO-RCT-PRF.SEED
READ-RCT-PRF.SEED
ADM-RCT-PRF.SEED
ELO-RST-PRF.SEED
READ-RST-PRF.SEED
ADM-RSTPRF.SEED

Table 58: User Profiles for National Gateways Access to Central Services Web interface

The following profiles have to be defined on DG TAXUD Gateways to control the access to SEED Web Services:

ccnUserProfileId
WS-ELO-LCT-PRF.SEED
WS-READ-LCT-PRF.SEED

Table 59: User Profiles for DG TAXUD Gateway Access to Central Services Web Services

The following profiles have to be defined on DG TAXUD Gateways to control the access to the SEED Web interface:

ccnUserProfileId
ELO-LCT-PRF.SEED
READ-LCT-PRF.SEED
ADM-LCT-PRF.SEED

Table 60: User Profiles for DG TAXUD Gateway Access to Central Services Web interface

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

XI.II.3 CCN/HTTP Configuration Procedure

The overall procedure for the configuration of CCN/HTTP is resumed by the following points:

1. The Central Application Designer creates configuration forms for the Central Services application;
2. The Central Application Designer sends the forms to the CCN-TC;
3. The CCN-TC configures the DG TAXUD and National Gateways.

For the MSA that opts for the additional WS-security, the overall procedure for the configuration of WS-security is resumed by the following points:

1. The Local Security Officer of the MSA sends its NDEA public Key to EMCS-CO;
2. The Central Security Officer of EMCS-CO replies with the Central Services public key.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Sub-Section XI.III Web Service Standards

Any MSA that interacts with web services of Central Services should conform to the following web service standards: SOAP, HTTP, XML-RPC, WSDL and WS-Security. This section specifies which version of the standard has to be supported by any MSA. It also gives a brief introduction to these standards.

XI.III.1 HTTP

The version of HTTP supported is HTTP/1.1.

XI.III.2 SOAP

SOAP is a format for *transmitting data* and web service *invocation calls* between two SOAP nodes - a SOAP sender and SOAP receiver, as shown below:

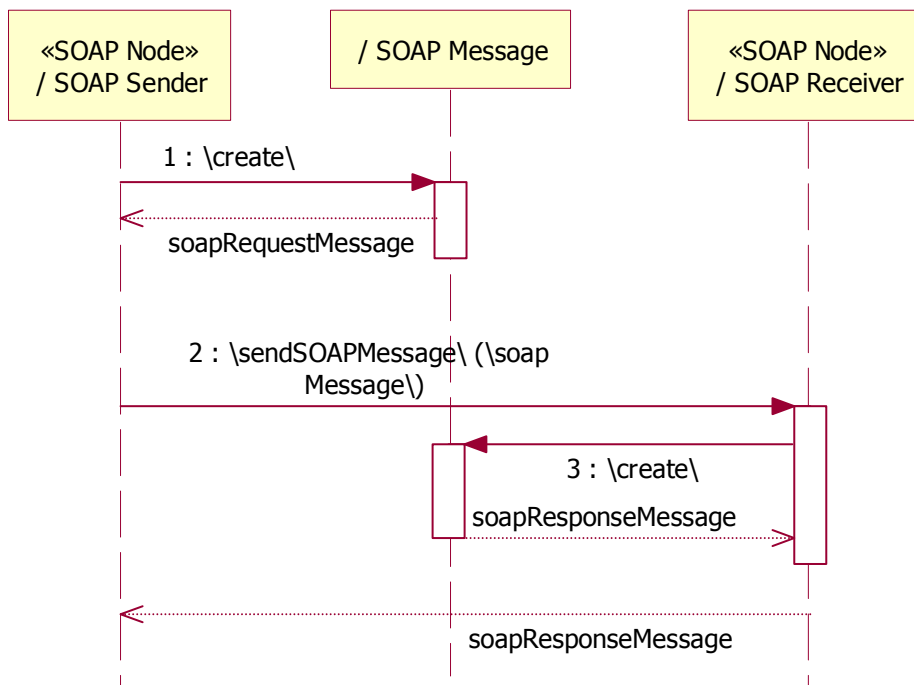


Figure 237: Generic SOAP Interaction

In the basic form of interaction, SOAP is a synchronous, one-way communication between a SOAP Sender and SOAP Receiver. However, as the diagram shows these one-way interactions may be combined in different SOAP message exchange patterns. The SOAP Receiver may, therefore, also perform the role of a SOAP Sender by sending a SOAP response to a SOAP request.

The versions of SOAP supported by the web services of Central Services are:

- SOAP 1.1;
- SOAP 1.2.

It is recommended that any MSA that will be a client of the web services conforms to the SOAP 1.2 standard. This is primarily because SOAP 1.2 has clearer semantics, in addition to a number of significant syntax changes, for example in relation to HTTP binding.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

XI.III.2.1 Encoding Style

SOAP supports two encoding styles for invocation of web services: Document Literal or SOAP RPC (Remote Procedure Call) encoding:

- In the Document Literal encoding style, a XML message is wrapped inside the SOAP envelope;
- The SOAP RPC encoding style defines a uniform representation of a RPC-like request/response interaction.

The Document Literal style is more flexible in that custom XML messages may be defined to be wrapped by the SOAP Envelope. In the context of EMCS, this means the existing message format used for XML/CSI interaction may be re-used for SOAP/HTTP invocation of the web services.

It is mandatory that any MSA that will be a client of the web services conforms to the SOAP Document Literal encoding style.

XI.III.3 WSDL

WSDL (Web Services Description Language) defines “a XML grammar for describing network services as collections of communication endpoints capable of exchanging messages”. WSDL *describes* web service operations, parameters and how a client connects to a web service.

The version of WSDL supported by the web services of Central Services is WSDL 1.1. The WSDL for these web services are included in the Appendix I.

XI.III.4 UDDI

UDDI (Universal Description, Discovery and Integration) is a standard for description and discovery of registered web services in a known repository.

Central Services will *not* support dynamic discovery of web services. That is, the WSDL for the web services will not be published in a UDDI repository for discovery by MSA clients. This is because the UDDI registry would have to be secured and client invocation would be slower if the WSDL for these web services is stored in an UDDI repository.

XI.III.5 WS-Security

The specific features of WS-Security and the implications for any MSA wishing to improve security are further defined in the Sub-Section XI.VI “Web Service Security” below.

Sub-Section XI.IV Recommended Usage

This section explains the requirements for a MSA that communicates with the web services over the SOAP/HTTP transport.

The first chapter details the SOAP message structure.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

The second chapter discusses the asynchronous modes of interaction between a MSA and the web services of Central Services. Specifically, this sub-section details how a MSA and Central Services may interact asynchronously through a SOAP conversation, and how the MSA should handle exceptions thrown by a web service.

XI.IV.1 SOAP Message Structure

The structure of a SOAP 1.2 message that is transmitted between SOAP nodes (web service client and receiver) is a SOAP Envelope that wraps the Central Services XML message format (IEs). The next sections specifies this SOAP Envelope, details the relationship between this SOAP Envelope and the XML format, and gives examples of the SOAP message structure that the MSA client must send to the web services.

XI.IV.1.1 SOAP 1.2 Envelope

The figure below illustrates the SOAP 1.2 Envelope:

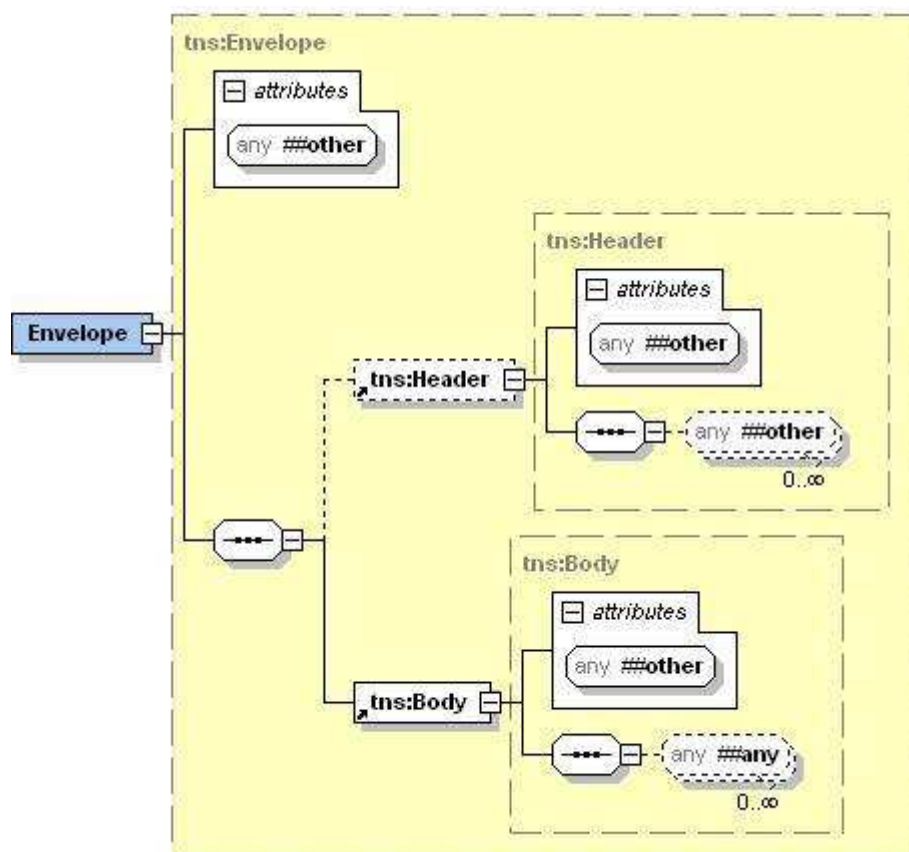


Figure 238: SOAP 1.2 Envelope

The SOAP Envelope is composed of a Header and a Body element. Both elements are extensible as the XML Schema for the envelope specifies that both the Header and Body elements may contain any other element. For example, this is expressed in XML Schema through the use of the `<xs:any>` type, as shown in the following excerpt for the `<tns:Body>` element:

```
<xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

This means any element from any namespace may be contained by the SOAP Envelope body schema in an instance of a SOAP message. It is this flexibility that makes the Document encoding style possible, so that XML messages specific to Central Services can be contained by the SOAP body element.

XI.IV.1.2 SOAP Body

The relationship between the SOAP Envelope and Central Services XML is dependent on the mode - asynchronous or synchronous - of message exchange. For the synchronous message exchanges the messages are contained in the body element of the SOAP Envelope. For the asynchronous message exchanges the messages are contained as attachments following the W3C recommendations, XML-binary Optimized Packaging (XOP) [S19] and SOAP Message Transmission Optimisation Mechanism [S20].

In the Document Literal encoding style, the operation invoked appears as an element contained by the <SOAP-ENV:Body> element. For example, the XML shown below is an excerpt from a SOAP request to the dissemination of SEED data operation of the 'MaintainEntity' web service:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
  MaintainEntity Service
  Operation: startEntityAction
  Attachment: IE713:C_QRO_DAT
-->
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <StartHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <startEntityAction
      xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The parameters of the operation are contained by the <startAction> element. As shown above, the operation element also contains the namespace declarations for the custom Central Services XML Schemas such as

```
<startEntityAction xmlns="http://emcs.dgtaxud.ec/webservice/types"/>
```

which identifies the Central Services excise domain schema.

MSA clients of Central Services Web Services should respect this style of encoding of SOAP requests. This is usually achieved through the use of developments tools, such as Microsoft Visual Studio.NET, that enable request creation dynamically given a web service WSDL file.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

XI.IV.2 SOAP Message Exchange Patterns

The Central Services supports the SOAP conversation specification, which is a set of SOAP headers, a protocol, and WSDL definitions designed to support asynchronous and conversational messaging over SOAP/HTTP.

All Central Services Web Services support asynchronous conversations. This is because they must support bulk validation, modification and download of multiples excise entities and such operations have the potential to take longer than the timeout for a HTTP request.

XI.IV.2.1 Asynchronous Message Exchange

In asynchronous exchange of messages, it is necessary for the SOAP client and server (or web service) to manage long-lived conversations over a period of time. To enable subsequent communication between SOAP nodes, a unique identifier or “conversation id” should be passed between the nodes when the conversation starts. Central Services uses a SOAP extension called the SOAP Conversation Protocol 1.0 to embed this identifier in the header element of the SOAP Envelope.

This imposes two requirements on any MSA that will interact with the asynchronous web services, specifically:

- A MSA that interacts with the asynchronous web services should supply a conversation id that is globally unique, e.g., a Globally Unique Identifier (GUID), a Uniform Resource Locator (URL) or a secure random number. This conversation identifier should be embedded in the SOAP Envelope Header element in every request to invoke an asynchronous operation;
- The MSA must poll the web service to get the current status of the requested entity action.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

The general sequence of interaction for asynchronous message exchange with a web service is shown in the figure below:

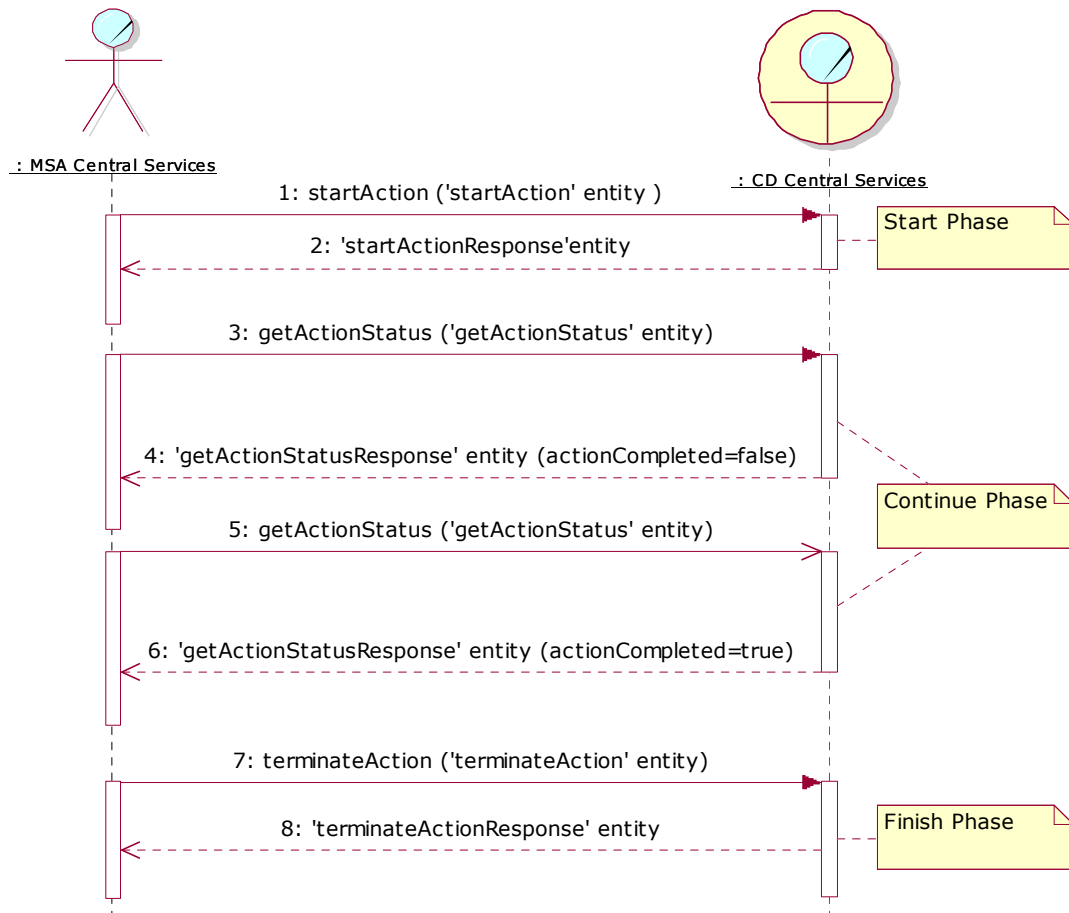


Figure 239: SOAP Conversation Phases

As the diagram shows, there are three phases in the SOAP conversation: start, continue and finish. In the start conversation phase, the MSA should send a 'StartHeader' entity (element) inside the SOAP Envelope Header element in the request to invoke the 'startAction' operation. The 'startHeader' entity contains the conversation identifier (conversationID) generated by the MSA client:

```

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Header>
    <StartHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </StartHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
  
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

In the continued conversation phase, the MSA must send a 'ContinueHeader' in any SOAP message to invoke the polling operation 'getActionStatus'. The 'ContinueHeader' should contain the same 'conversationID' value originally sent in the 'StartHeader' element.

For example:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Header>
    <ContinueHeader
      xmlns="http://www.openuri.org/2002/04/soap/conversation/">
      <conversationID>9E6EEAE9-E64E-40f2-85FF-
A2CD8E35F6D0</conversationID>
    </ContinueHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    ...
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The polling operation may be invoked multiple times, each time with the same 'ContinueHeader'.

To terminate a conversation, the MSA must invoke the 'terminateAction' operation of the web service. In the SOAP message request to invoke 'terminateAction', the same 'ContinueHeader' *should also be sent* in the SOAP Envelope Header: in the chosen SOAP Conversation protocol, there is no equivalent of a stop or terminate header.

XI.IV.2.1.1 Conversation Lifetime

The duration of a SOAP Conversation is not unlimited. There are two specific attributes that control the lifetime of a SOAP Conversation. These are presented in the table below:

Attribute	Value	Type
Max-age	24	Hours
Max-idle-time	1	Hour

Table 61: SOAP Conversation Lifetime Attributes

The "max-age" attribute controls the overall lifetime of a conversation. In Central Services, the "max-age" of the web services is set to 24 hours, after which the conversation terminates.

The "max-idle-time" attribute controls the time between invocations of a web service for the same conversation. The "max-idle-time" of the SOAP conversation is set to one hour, after which the conversation terminates.

If any operation is invoked by the MSA after the conversation has ceased, the server throws a SOAP Fault (see XI.IV.2.2 "Exception Handling" below) back to the MSA client. The SOAP Fault specifies that the SOAP Conversation identified by the 'conversationID' sent in a 'ContinueHeader' has terminated.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

At this point, the MSA may start a new conversation by sending a 'StartHeader' containing a new 'conversationID' with the SOAP request to invoke the 'startAction' operation. Once the new conversation has started, the MSA may call the polling operation 'getActionStatus' to get the result of the entity action.

XI.IV.2.2 Exception Handling

The MSA client should be able to handle exceptions thrown by a web service of the Central Services. There are three kinds of exceptions that are possible:

- System exceptions occur when there is a network timeout or the web service is unavailable and are generated by the XML-RPC module of the MSA. The management of System exceptions is left to the discretion of the MSA implementation;
- Service exceptions that are application specific and generated by a web service if some parameters in a SOAP request are invalid or, alternatively, if the internal state of the web service is inconsistent. Service exceptions are returned to the MSA client as SOAP Faults.
- Functional Errors IE906, IE917

XI.IV.2.2.1 System Exceptions

A MSA invoking a web service operation should only do so in the context of an exception handling mechanism. In response to a system exception, a possible strategy for the MSA would be to retry the request, for a finite number of times (to be defined by MSA). In addition, it may be possible for the MSA to be configured with a wait period value, so that the MSA will effectively wait a period before retrying the SOAP request. In the case where the unavailability of the web service is temporary, a subsequent invocation of the web service may be successful.

Should the web service still be unavailable after all retry attempts have failed, the MSA has no option but to log the System exception, alert the Local System Administrator and return.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

XI.IV.2.2.2 Service Exceptions

The figure below shows a possible strategy for a NDEA response to a Service exception being thrown by a Central Services web service:

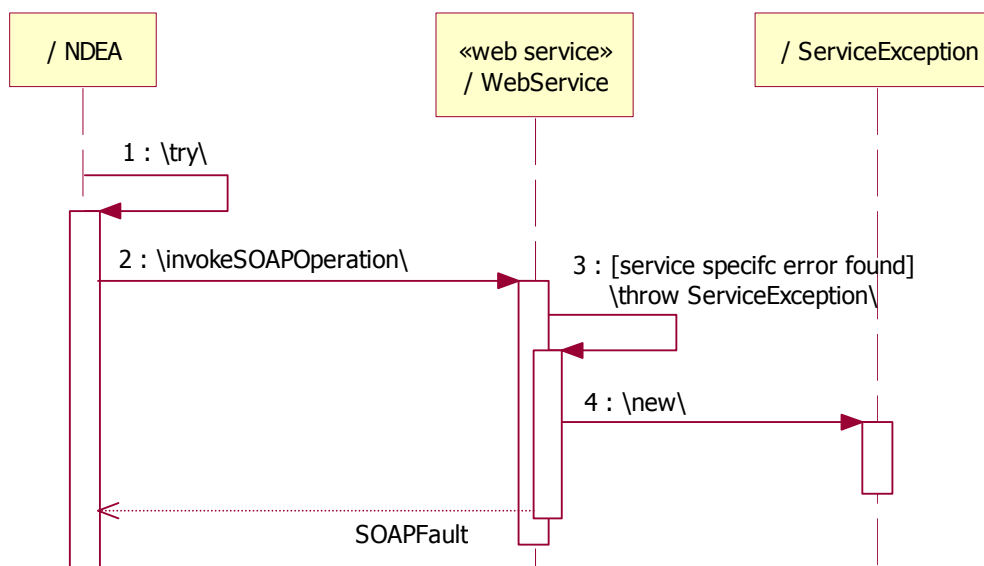


Figure 240: MSA Response to Service Exception.

As with the System exception, NDEA invoking a Central Services web service operation should only do so with a context of an exception handling mechanism. Should a service specific exception be found (e.g. if parameter values are invalid or the Central Services Web Service being in an inconsistent state to process the SOAP request), the web service throws a service exception back to the MSA client.

For instance, the “UpdateSEED” web service defines an operation to disseminate the updates which throws an “UpdateRejectedException” if there is a semantic problem with any data passed as a parameter.

A set of SOAP faults codes can be defined in Central Services described as following:

VersionMismatch: The processing party found an invalid namespace for the SOAP Envelope element

MustUnderstand: An immediate child element of the SOAP Header element that was either not understood or not obeyed by the processing party contained a SOAP mustUnderstand attribute with a value of "1"

Client: The Client class of errors indicate that the message was incorrectly formed or did not contain the appropriate information in order to succeed. For example, the message could lack the proper authentication or payment information. It is generally an indication that the message should not be resent without change.

Server: The Server class of errors indicate that the message could not be processed for reasons not directly attributable to the contents of the message itself but rather to the processing of the message. For example, processing could include communicating with an upstream processor, which didn't respond. The message may succeed at a later point in time.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

XI.IV.2.2.3 Functional Errors

Messages for error reporting are IE906 and IE917. The IE917 is used for reporting XML formatting errors, while Functional IE906 is used for reporting functional errors (e.g. violation of Information Exchange building rules). Note that the error reporting messages will be applicable to all scenarios of Common Domain Central Services

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Sub-Section XI.V Web Service Transactions

Every invocation of a web service operation initiates an implicit transaction. For the MSA client, this has several implications, which can be summarised as follows:

- If the invocation is successful i.e. no SOAP Fault or an IE906/IE917 is returned to the MSA, the transaction has been committed;
- For the web services, which perform update operations on entities (i.e. MaintainSEED, UpdateAvailability), if the operation succeeds then *all* the actions specified in the SOAP request have succeeded.
- If a web service throws an exception to the MSA in the form of a SOAP Fault or IE906/IE917, then the transaction has been rolled back;
- For the web service, if the operation fails then *all* the actions specified in the SOAP request have failed.

XI.V.1 Asynchronous Web Services

Asynchronous web services have additional transactional behaviour, depending on the phase of the conversation lifetime:

- In the start phase of a conversation, if an exception is thrown then the conversation is never started, so that the MSA must try to start a new conversation as described in the discussion of the XI.IV.2.1.1 Conversation Lifetime above;
- If an exception is thrown during the “continue” or “finish” phases of a conversation, the conversational state is not updated. That is, the conversation remains at the previous state and the MSA should retry the operation that failed. By implication, if the exception is thrown during the “finish” phase of the conversation, the conversation is not terminated.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

Sub-Section XI.VI Web Service Security

Central Services Web Services may be secured on two levels: transport and message:

- Transport level security concerns authentication and authorisation of HTTP transport between the MSA and web services;
- An optional additional level of security concerns securing the actual content of the SOAP message exchanged. A specific OASIS specification called “WS-Security” addresses such message-level security.

XI.VI.1 HTTP Transport Security

The standard way of securing HTTP transport is the use of the Secure Socket Layer (SSL). SSL can be either one-way or two-way:

- In one-way SSL, the identity of the server is confirmed through the presentation of a certificate to the client and communication between client and server is encrypted;
- In two-way SSL, both the client and server are required to present a certificate during an exchange that precedes the establishment of a secure SSL connection.

Transport-level security in Central Services is achieved through the use of “one-way SSL with form and basic Authentication”. The reasons for this choice are that one-way SSL is sufficient to ensure the integrity of the communication between client and server through encryption of the data and that two-way SSL is more complex and requires the MSA to programmatically send a digital certificate to the server.

In order to achieve authentication of the central MSA to the Central Services web service, it is necessary to supplement one-way SSL with either form-based or basic authentication.

Central Services Web Services will support both:

- Form-based authentication for interactions through the CCN Network. This function is implemented by CCN;
- Basic authentication is used for access from the Commission intranet.

XI.VI.1.1 CCN Authentication and Authorisation

Both user and application authentication will be needed for replacing the CCN Ticket Login Mechanism with the CCN Lightweight Session Cookie mentioned in “CCN Intranet Services - Programmer’s Guide” [R10]. The user authentication will be needed for the MSA user which supplies the Session ID to Central Services Application and the Application authentication for the relevant Server (CSRD) to get the profiles associated with the user.

Sub-Section XI.VII CCN user HTTP authentication

The authentication mechanism is activated each time access is requested to a web page defined as “protected access” in CCN. It is typically used when using a HTML browser.

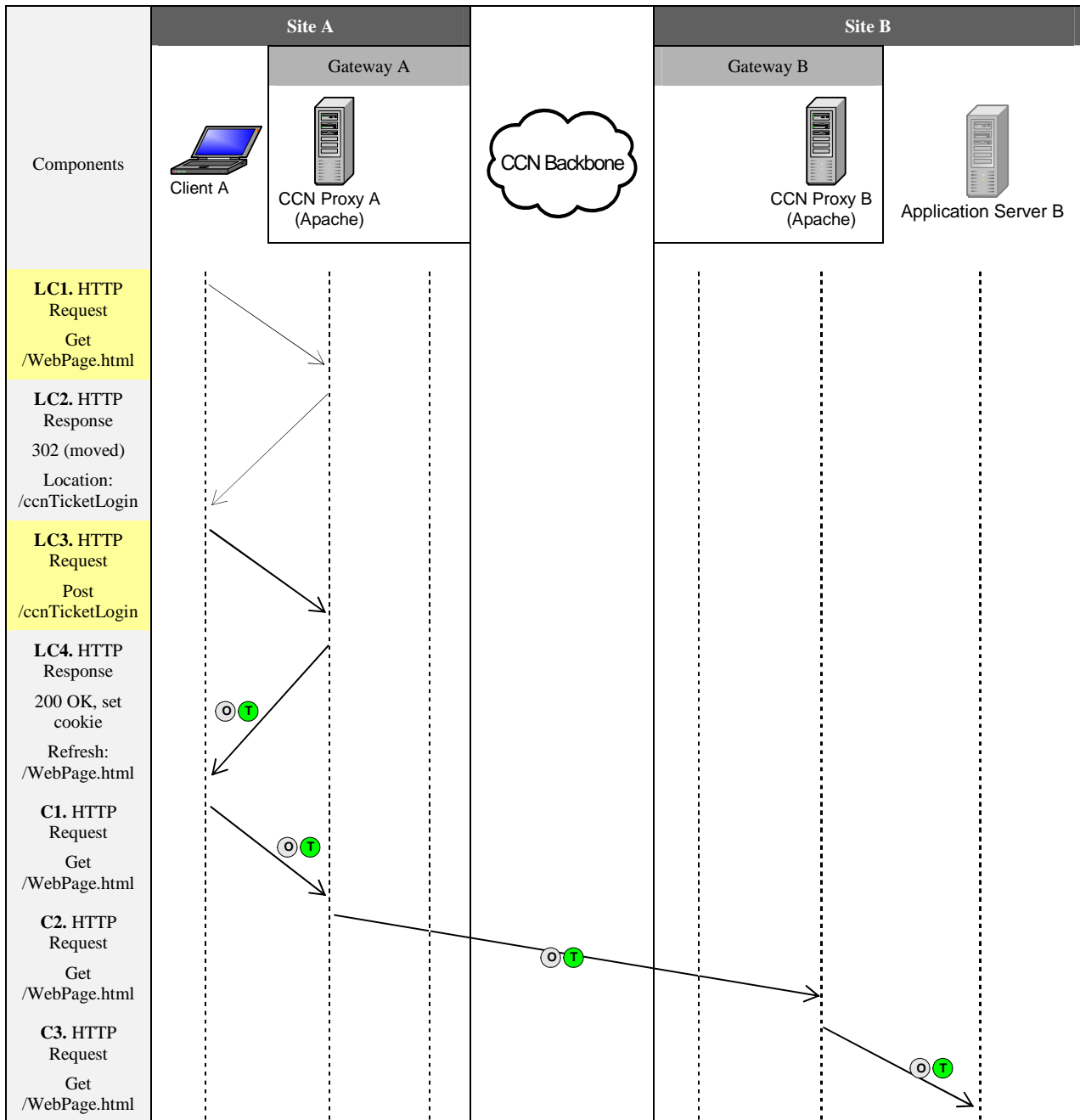


Figure 241: Http flow (user part)

LC1. The client browser sends a ‘HTTP get request’ for the page ‘/WebPage.html’

LC2. Because the URL is not registered as being freely accessed (in the Gateway directory) and there is no CCN cookie attached to the flow, the Apache server returns a 302 (moved), and the location of the ccnTicketLogin is given back to the browser

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XI - Transport of Messages via SOAP/HTTP	

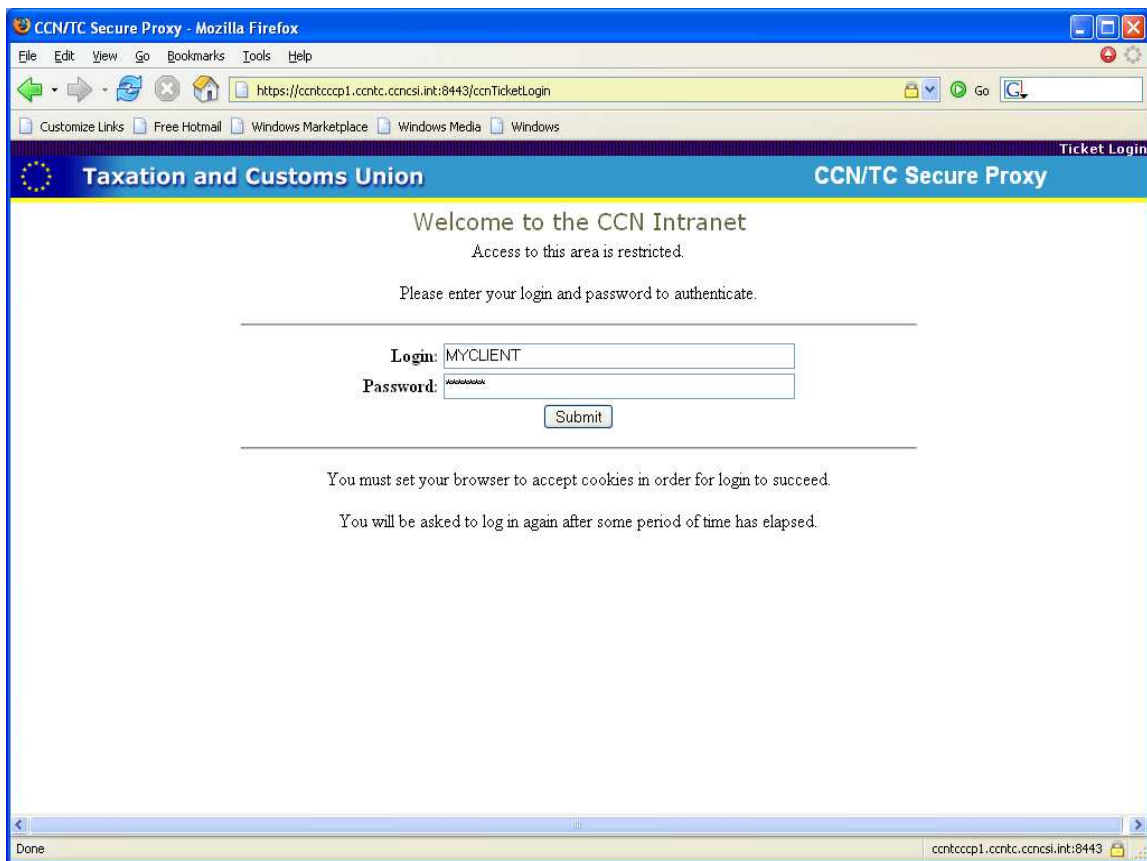


Figure 242: CCN Intranet user login screen

LC3. The client browser sends a ‘HTTP post for /ccnTicketLogin’ (Ticket handler) with the user and the password.

LC4. If the user is known in CCN and its password is correct, cookies will be built and attached to the flow.

C1. The client refreshes his ‘request to get /WebPage.html’. Two cookies are attached to the communication. The grey one is the old cookie (O), the green one (T) is the ccnSession lightweight ticket cookie. This cookie only contains a session ID (see §8.2). Relevant information about the user is stored on the gateway.

This session is verified by the ticket-handler. Based on user information linked to the session ID, access is granted or refused to the requested URL.

C2. Since the ‘CCN Proxy A’ is not the ‘request destination gateway’, the request is sent to the Proxy B (destination).

C3. The cookies are verified by the ticket-handler (MD5 key) (please refer to CCN Intranet Services - Programmer’s Guide [R10]). The ccnSession cookie is checked. And the request is sent to the destination (Application server).

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XII - Application authentication intranet services	

Section XII APPLICATION AUTHENTICATION INTRANET SERVICES

This section describes the services which provide HTTP interfaces to the web applications in order to **login** (ccnServerLogin) into CCN and **logout** (ccnServerLogout) from CCN.

Sub-Section XII.I ccnServerLogin service

The table below describes the syntax of this service:

This service provides a HTTP interface to allow a web-application to authenticate into CCN.	
<u>Request</u>	
<i>The request consists sending a POST of the following url:</i>	
<i>https://[Gateway].[site].ccncsi.int:8443/ccnServerLogin</i>	
Input	<ul style="list-style-type: none"> • User
	<ul style="list-style-type: none"> • Password
<u>Response</u>	
<i>The response consists in a XML buffer embedding the following information:</i>	
Output	<ul style="list-style-type: none"> • TICKET_RESPONSE
<u>Cookies set in response</u>	
<ul style="list-style-type: none"> • ccnSession • Ticket (obsolete but still created) 	

Table 62: ccnServerLogin syntax and description

XII.I.1 DTD of the response

```
<?xml version='1.0'?>
<!DOCTYPE TICKET_RESPONSE [
  <!ELEMENT TICKET_RESPONSE (#PCDATA)>
  <!ELEMENT TICKET_RESPONSE ( ERROR? )>
  <!ATTLIST TICKET_RESPONSE name (TICKET_RESP) #REQUIRED
                                returncode (SUCCESS|FAILURE) #REQUIRED
                                ccnSessionId CDATA #IMPLIED >
  <!ELEMENT ERROR (#PCDATA)>
  <!ATTLIST ERROR description CDATA #REQUIRED >
]>
```

Table 63: ccnServerLogin response's DTD

XII.I.2 Example of request

```
POST https://ccntcccpl.ccntc.ccncsi.int:8443/ccnServerLogin
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XII - Application authentication intranet services	

```
Query data:
user=TESTUSR&password=testPassword
```

Table 64: ccnServerLogin request example

XII.I.3 Example of successful response

```
HTTP/1.1 200 OK
Set-Cookie: Ticket =
ip&209.28.97.202&time&1123576862&user&TESTUSR&profiles&ADM2G-ADM2G-
PRF.ADM2G WEBMON-
PRF.VIES&site&CCN.TC&hash&85097673fbbacdc571e6c4021afc8317&expires&480
; path=/; domain=.ccncsi.int
Set-Cookie: ccnSession = id&c36cbe6fa14e0a08e8e06683f157fd6e; path=/;
domain=.ccncsi.int

<?xml version='1.0'?><TICKET_RESPONSE name="TICKET_RESP"
return_code="SUCCESS"
ccnSessionId="c36cbe6fa14e0a08e8e06683f157fd6e" />
```

Table 65: ccnServerLogin success response example

XII.I.4 Example of unsuccessful response

```
HTTP/1.1 200 OK

<?xml version='1.0'?><TICKET_RESPONSE name="ERROR"
return_code="FAILURE"><ERROR description="ERROR: The user CCNADM-
USRxxx cannot be authenticated, error in
srm_get_ext_sec_attributes" /></TICKET_RESPONSE>
```

Table 66: ccnServerLogin error response example

Sub-Section XII.II ccnServerLogout service

This service provides a HTTP interface to allow a web-application to logout from CCN.	
<u>Request</u>	
<i>The request consists in calling a GET of the following url:</i>	
<i>https://[Gateway].[site].ccncsi.int:8443/ccnServerLogout</i>	
<u>Cookie needed in request</u>	
<ul style="list-style-type: none"> ccnSession (this cookie is the set in the ccnServerLogin) 	
<u>Response</u>	
<i>The response consists in a XML buffer embedding the following information:</i>	
Output	<ul style="list-style-type: none"> TICKET_RESPONSE

Table 67: ccnServerLogout syntax and description

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XII - Application authentication intranet services	

XII.II.1 DTD of the response

```
<?xml version='1.0'?>
<!DOCTYPE TICKET_RESPONSE [
  <!ELEMENT TICKET_RESPONSE (#PCDATA)>
  <!ELEMENT TICKET_RESPONSE ( ERROR? )>
  <!ATTLIST TICKET_RESPONSE name (TICKET_RESP) #REQUIRED
    returncode (SUCCESS|FAILURE) #REQUIRED
    ccnSessionId CDATA #IMPLIED >
  <!ELEMENT ERROR (#PCDATA)>
  <!ATTLIST ERROR description CDATA #REQUIRED >
]>
```

Table 68: ccnServerLogout response's DTD

XII.II.2 Example of request

```
GET https://ccntcccpl.ccntc.ccncsi.int:8443/ccnServerLogout

Cookie Data:
Ticket = ip&209.28.97.202&time&1123576862&user&TESTUSR&profiles&ADM2G-ADM2G-PRF.ADM2G WEBMON-PRF.VIES&site&CCN.TC&hash&85097673fbbadc571e6c4021afc8317&expires&480; path=/; domain=.ccncsi.int; ccnSession = id&c36cbe6fa14e0a08e8e06683f157fd6e
```

Table 69: ccnServerLogout request example

XII.II.3 Example of successful response

```
HTTP/1.1 200 OK

<?xml version='1.0'?><TICKET_RESPONSE name="TICKET_RESP"
return_code="SUCCESS"/>
```

Table 70: ccnServerLogout success response example

XII.II.4 Example of unsuccessful response

```
HTTP/1.1 200 OK

<?xml version='1.0'?><TICKET_RESPONSE name="ERROR"
return_code="FAILURE"><ERROR description="ERROR: The information about
session c36cbe6fa14e0a08e8e06683f1zzzzzz cannot be
retrieved"/></TICKET_RESPONSE>
```

Table 71: ccnServerLogout error response example

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XII - Application authentication intranet services	

XII.II.5 WS-Security

WS-Security offers ‘message-level’ security, which supplements ‘transport level’ security by addressing its weaknesses:

- Transport level security is only ‘*point-to-point*’ - SSL only encrypts the connection between the MSA HTTP client and Central Services. If the HTTP request is proxied via an intermediary (for instance, via the ND and CD Apache Servers), the intermediary receives the SOAP message in clear text. In contrast, ‘message-level’ security offers ‘*end-to-end*’ security: WS-Security ensures the *integrity* of a SOAP message. That is, the SOAP message is not available in clear text to any intermediary, so it is not possible alter the message content;
- With Transport level security, if a HTTP request is proxied, the originator of the request (MSA) may no longer be known to the receiver (Central Service). In contrast, with message level security it is possible to authenticate the identity of the sender (MSA) through the use of XML Digital Signatures;
- SSL encrypts everything; where as, in message level security with XML Encryption it is possible to protect the *confidentiality* of a SOAP message by encrypting only those specific parts of SOAP message that contain confidential information.

Central Services supports the OASIS WS-Security standard. WS-Security is defined in the OASIS standard specification.

The SOAP extensions take the form of a security element which has the following structure:

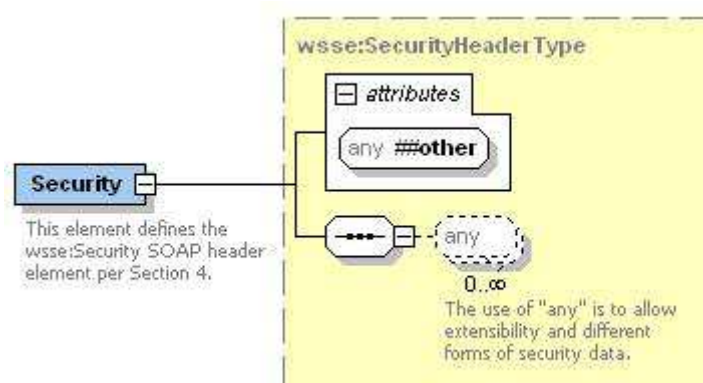


Figure 243: WSSE Security Structure

In an instance of a SOAP request, the `<wsse:Security>` element is embedded in the Header element of the SOAP Envelope in both SOAP requests and responses:

```
<SOAP-ENV:Header>
  <wsse:Security
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/07/secext "
    SOAP-ENV:mustUnderstand="1">
    ...
  </wsse:Security>
</SOAP-ENV:Header>
```

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Section XII - Application authentication intranet services	

As shown in the previous figure, the `<wsse:Security>` element is extensible and can contain different kinds of security data. That is, the `<wsse:Security>` element may contain a `<wsse:UsernameToken>` or `<wsse:BinarySecurityToken>`. In addition, it may contain a `<dsig:Signature>` digital signature element. Finally, it may contain a `<xenc:EncryptedKey>` element to describe a key used to encrypt the content of a SOAP message. The diagram below shows the actual structure of a `<wsse:Security>` element in an instance of a SOAP request or response:

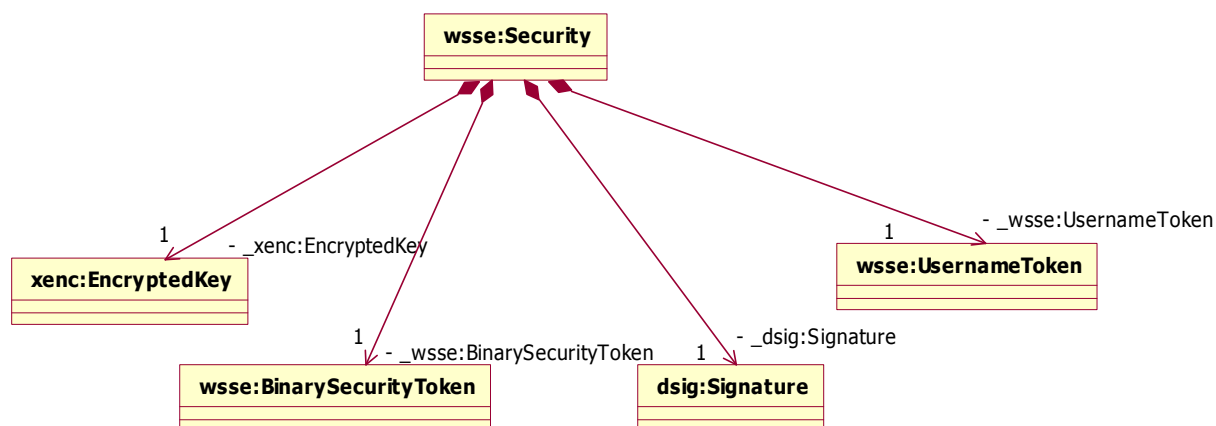


Figure 244: SOAP Request/Response Elements Contained by the WSSE Security Structure

In addition to the `<wsse:Security>` element contained in the SOAP Envelope header, if encryption is to be used to ensure message confidentiality, the Body of the SOAP Envelope may contain encrypted XML data according to an encryption scheme defined within the `<wsse:Security>` element.

For example:

```
<SOAP-ENV:Body Id="Id-M3yXsNQfLaNYOOSpoofv1mZg">
  <xenc:EncryptedData
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Id-VDFRJSxZTq4eE5zWJ5n/N+Ex"
    Type="http://www.w3.org/2001/04/xmlenc#Element">
    ...
  </xenc:EncryptedData>
</SOAP-ENV:Body>
```

Currently 2 fields named X509Certificate and SignatureValue in the Format:Based64 Binary will be included in every message in order to ensure WS-security applicability.

However, the OASIS standard documentation should remain the definitive source of information on WS-Security.

EMCS Phase 3	ECP3-FITSDEV3-SC01-DDNEA
DDNEA for EMCS Phase 3	VER.: 1.76-EN
Appendices	

APPENDICES

Appendix A: MESSAGE SCOPE

Appendix B: CODELISTS

Appendix C: EMCS CORRELATION TABLE

Appendix D: TECHNICAL MESSAGE STRUCTURE

Appendix E: XML MAPPING

Appendix F: DATA GROUPS & TRANSACTION HIERARCHY

Appendix G: DATA ITEMS

Appendix H: DIRECTORY WITH XML SCHEMAS (XSDs)

Appendix I: DIRECTORY WITH WEB SERVICE INTERFACE DEFINITIONS (WSDLs)

Appendix J: BUSINESS RULES CATALOGUE

Appendix K: RULES AND CONDITIONS