COUNCIL OF
THE EUROPEAN UNION

Brussels, 9 June 2011

11275/11

LIMITE

COPEN 151
EJUSTICE 52
JURINFO 42

NOTE

from:	General Secretariat of the Council
to:	Delegations
Subject:	ECRIS Detailed Technical Specifications

Delegations will find in the Annex the revised text of the ECRIS Detailed Technical Specifications as resulting from the observations submitted by the Member States and examined during the discussions at the Working Party on Cooperation in Criminal Matters, which met on 19 May 2011. Changes are indicated by underlined text as against the initial ECRIS Detailed Technical Specifications.

ANNEX

European Commission – DG Justice

iLICONN Consortium (Bilbomatica – Intrasoft – Unisys)

ECRIS Technical Specifications

Detailed Technical Specifications

Document Information

AUTHOR	iLICONN – Intrasoft International S.A.
OWNER	European Commission – DG Justice
ISSUE DATE	28/02/2011
VERSION	1.4
APPROVAL STATUS	Final

Authors

NAME	ACRONYM	ORGANISATION	ROLE
Nicholas YIALELIS	NYI	iLICONN – Intrasoft International S.A.	Manager Reviewer
Panos ATHANASIOU	PAT	iLICONN – Intrasoft International S.A.	Main Author
Ludovic COLACINO DIAS	LCO	iLICONN – Intrasoft International S.A.	Contributor Reviewer

Document History

VERSION	DATE	AUTHOR	DESCRIPTION
0.1	29/10/2010	LCO	First draft
0.2	03/11/2010	PAT	Updates on all sections
0.3	05/11/2010	PAT	Updates on all sections
0.4	07/11/2010	PAT	Minor updates in various sections
0.5	10/11/2010	LCO	Revision
0.6	24/11/2010	PAT	Update for implementing comments
0.7	25/11/2010	LCO	Revision for delivery of author's position
0.8	17/12/2010	PAT	Implementation of comments and changes agreed during the Expert Group Review meeting 01-Dec-2010 and COPEN meeting 10-Dec-2010
0.9	20/12/2010	LCO	Revision before final delivery
1.0	22/12/2010	LCO	Final delivery
1.1	23/02/2011	DCO	Implementation of comments and changes agreed during the Expert Group Review meeting 16-Feb-2011
1.2	24/02/2011	PAT	Implementation of comments and changes agreed during the Expert Group Review meeting 16-Feb-2011
1.3	28/02/2011	PAT	Minor corrections
1.4	28/02/2011	LCO	Final revision
1.5	01/03/2011	DCO	Minor corrections after revision by DG JUST

TABLE OF CONTENTS

1 DOCUMENT

1.1 Purpose

This document is a formal product of the *ECRIS Technical Specifications* project for the European Commission – DG Justice and produced by the iLICONN Consortium.

The main purpose of this document is to describe the technical artefacts that provide the detailed technical specifications of the ECRIS data exchanges. It presents the logic, rules and detailed content of the technical artefacts constituting the *ECRIS Technical Specifications* and aims at providing all necessary information and guidelines for the developments teams that will need to implement these technical specifications.

This document assumes that the readers have a good and detailed knowledge and understanding of the following elements:

- The ECRIS legal basis (Council Framework Decision 2009/315/JHA and Council Decision 2009/316/JHA)
- The "ECRIS Technical Specifications – Inception Report" document
- The "ECRIS Technical Specifications – Technical Architecture" document
- The "ECRIS Technical Specifications – Security Analysis" document
- The "ECRIS Technical Specifications – Business Analysis" document
- Detailed knowledge and understanding on IT-technical level of *Web Services*, *XSD* and *XML*

1.2 Scope

This document provides all necessary background information so as to gain a deep understanding of the technical artefacts that define the ECRIS *Detailed Technical Specifications*. In particular, this document provides:

- the general descriptions of the WSDL, XSD and XML files provided
- the general logic, rules and guidelines that have been followed for designing and defining these technical artefacts
- detailed descriptions of the design choices
- UML class diagrams depicting in a logical form the relations between the XSD entities that are defined in the technical artefacts
- UML sequence diagrams illustrating the sequences in which the *web service* operations are meant to be used
- roles of the involved stakeholders for the management and future evolutions of the technical artefacts
- the minimum set of technical and logical validation rules that must be implemented in the ECRIS software
- recommendations and guidelines for the implementers of the detailed technical specifications

This document does not provide any other information than what has been stated above, and in particular it does not include:

- a copy of the content of the *WSDL*, *XSD* and *XML* files so as to avoid redundancy; it is expected that the reader opens these technical files into appropriate reading/editing software in addition to reading this document
- the overall technical system architecture (please refer to the "Technical Architecture" document)
- the functional and business concepts (please refer to the "Business Analysis" document)
- the IT-security considerations related to the ECRIS data exchanges (please refer to the "Security Analysis" document)

1.3 References

The following sources have been used as input for the elaboration of this "Technical Architecture" document:

[1] ECRIS Legal Basis – Council Framework Decision 2009/315/JHA

Council Framework Decision 2009/315/JHA of 26 February 2009 on the organisation and content of the exchange of information extracted from the criminal record between Member States (OJ L 93/23 of 07.04.2009)

[2] ECRIS Legal Basis – Council Decision 2009/316/JHA

Council Decision 2009/316/JHA of 6 April 2009 on the establishment of the European Criminal Records Information System (ECRIS) in application of Article 11 of Framework Decision 2009/315/JHA (OJ L 93/33 of 07.04.09)

[3] Network of Judicial Registers (NJR) – Technical References – version 1.3a (approved) of 13 March 2008

[4] Network of Judicial Registers (NJR) – Technical References – version 1.4 (draft) of 23 November 2009

[5] Network of Judicial Registers (NJR) – *XML* Listings – version 1.4 (final) of 01 July 2009

[6] NJR *WSDL* and *XML* Files v1.4.2 of 21 January 2009 (final)

"CommonTables_and_XML_rel1-4-2_20090121.zip" file containing:

- RegisterService-1.4.2.wsdl (version 1.4.2)
- common.xsd (version 1.4 of 18 December 2008)
- CommonTables-1.3.xsd (version 1.3)
- CommonTables-1.4.2.xml (version 1.4.2)
- error.xsd (version 1.4 of 02 November 2005)
- information.xsd (version 1.4 of 02 November 2005)
- notification.xsd (version 1.4 of 22 November 2005)
- receipt.xsd (version 1.4 of 02 November 2005)
- request.xsd (version 1.4 of 02 November 2005)

[7] NJR *WSDL* and *XML* Files v1.5 (draft)

- RegisterService-1.5.wsdl (draft version 1.5 of 11 August 2010)
- common.xsd (draft version 1.5 of 10 June 2010)
- CommonTables-1.5.xsd (draft version 1.5)
- CommonTables-1.5.xml (draft version 1.5.0)
- error.xsd (draft version 1.5 of 10 July 2010)
- information.xsd (draft version 1.5 of 10 July 2010)
- notification.xsd (draft version 1.5 of 10 July 2010)
- receipt.xsd (draft version 1.5 of 10 July 2010)
- request.xsd (draft version 1.5 of 10 July 2010)

[8] European Commission – DG Enterprise (2004): IDA Architecture Guidelines for Trans-European Telematics Networks for Administrations, version 7.1 of 13 February 2004 (and annexes)

[9] ECRIS Technical Specifications - Inception Report v1.02 of 22 October 2010

[10] ECRIS Technical Specifications – Glossary v1.3 of 07 November 2010

[11] ECRIS Technical Specifications – Technical Architecture v1.0 of 22 October 2010

[12] ECRIS Technical Specifications – Security Analysis v1.0 of 22 October 2010

[13] ECRIS Technical Specifications – Business Analysis v1.4 of 28 February 2011

[14] ECRIS Technical Specifications – Logging, Monitoring and Statistics Analysis v1.5 of 01 March 2011

[15] ECRIS Technical Specifications – Common Reference Tables v0.18 of 25 February 2011

[16] The replies from the following Member States to the "Inception Phase Questionnaire" v0.05 (listed in alphabetical order):

> Austria (AT), Belgium (BE), the Czech Republic (CZ), Estonia (EE), Finland (FI), France (FR), Germany (DE), Greece (GR), Hungary (HU), Italy (IT), Lithuania (LT), Luxembourg (LU), the Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), the United Kingdom (UK)

[17] "SOA Design Patterns"

Thomas Erl - Prentice Hall/PearsonPTR (ISBN: 0136135161)

[18] "Web Service Contract Design and Versioning for SOA"
Thomas Erl, Anish Karmarkar, Priscilla Walmsley, Hugo Haas, Umit Yalcinalp, Canyang Kevin Liu, David Orchard, Andre Tost, James Pasley - Prentice Hall/PearsonPTR (ISBN: 013613517X)

[19] "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions"
Gregor Hohpe, Bobby Wolf - Addison-Wesley Professional
(ISBN-13: 978-0321200686)

[20] "Creating flexible and extensible *XML* schemas"
Ayesha Malik, 01 Oct 2002
http://www.ibm.com/developerworks/library/x-flexschema/

[21] Guide to Versioning *XML* Languages using *XML* Schema 1.1
David Orchard (2006 W3C)
http://www.w3.org/TR/xmlschema-guide2versioning/

[22] ECRIS-TCN Study: Final report of "Feasibility Study: Establishment of a European Index of Convicted Third Country Nationals", 11 June 2010

1.4 About this Document

1.4.1 Elaboration of this Document

This document has been drafted by the iLICONN staff based on the following input:

- The documents listed in the references above
- Information gathered during the preliminary on-site visits of the following Member States' central authorities:
  - 19-Jul-2010 / 30-Jul-2010: Belgium – Service Public Fédéral Justice – Service Casier Judiciaire Central
  - 26-Jul-2010 : France – Ministère de la Justice – Casier Judiciaire National
  - 29-Jul-2010 : Germany – Bundesamt für Justiz – Bundeszentralregister
  - 05-Aug-2010 : United Kingdom – Association of Chief Police Officers (ACPO) – ACPO Criminal Records Office (ACRO)
  - 09-Aug-2010 : Spain – Ministerio de Justicia – Registro central de penados y rebeldes
- The answers provided by the following Member States' central authorities to the questions defined in the "Inception Phase Questionnaire" document that has been sent out by the European Commission to all Member States' contact points on the 04th of August 2010 (listed in alphabetical order):

  > Austria (AT), Belgium (BE), the Czech Republic (CZ), Estonia (EE), Finland (FI), France (FR), Germany (DE), Greece (GR), Hungary (HU), Italy (IT), Lithuania (LT),

Luxembourg (LU), the Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), the United Kingdom (UK)

- Direct contacts and meetings with various experts (various experts from the European Commission, experts from the contractor currently developing the *NJR Reference Implementation* software but also other experts that have been involved in various studies and similar projects in the field of justice and cooperation in criminal matters).
- Concrete examples of NJR "notifications", "requests" and "information" messages provided by the following Member States: BE, FR, ES and UK.
- The 263 comments issued by the Member States and the European Commission on the previous version of this document until 19 November 2010.
- The discussions and agreements that have been reached during the Expert Group Review meeting on 01 December 2010 and COPEN Working Party meeting on 10 December 2010.
- The comments issued by the Member States and the European Commission on the previous version of this document.
- Conference calls held with experts from DE, ES, FR and PL between 7 and 9 February 2011 and direct e-mail contacts with several other Member States for discussing the aforementioned comments.
- The discussions and agreements that have been reached during the Expert Group Review meeting on 16 February 2011.

1.4.2 Understanding this Document

This document comes with a "Glossary" document that provides definitions for the specific terms that are used throughout the *ECRIS Technical Specifications* project. By convention, all words marked in italic in this document can be looked up in the "Glossary" document. The bold font is used for emphasising a specific term or part of a sentence. The underlines mark the text that has been added or modified since the last version while the strike-through marks the text that has been removed or replaced. In case of doubts about the exact meaning of a term, please consult first the "Glossary".

In addition, this document comes with the technical artefacts that constitute the *ECRIS Detailed Technical Specifications* to be implemented, in form of *WSDL*, *XSD* and *XML* files:

- ecris-service-v1.0.wsdl
- messages-v1.0.xsd
- domain-model-v1.0.xsd
- common-reference-tables-v1.0.xsd
- common-reference-tables-v1.0.xml
- commons-v1.0.xsd

Should you still have any doubts about the meaning of a specific sentence or paragraph, please do not hesitate to take direct contact with the following persons by telephone or via e-mail, at your best convenience:

Organisation:	European Commission – DG Justice – Criminal Law
Name:	Jaime LOPEZ-LOOSVELT
E-mail:	JUST-CRIMINAL-RECORD@ec.europa.eu
Telephone:	+32 (0)2.298.41.54

Organisation:	iLICONN Consortium – Intrasoft International S.A.
Name:	Ludovic COLACINO DIAS
E-mail:	ECRIS-Specs-PM.iLICONN@intrasoft-intl.com
Mobile:	+32 (0)498.30.25.55

1.4.3 Providing Comments

As described in the "Inception Report" document, all major deliverables produced by the iLICONN Consortium are undergoing a "Review Cycle" during which all EU Member States experts are invited to provide comments.

Commenting this document

Since the iLICONN staff needs to collect, compare and analyse the feedback from 27 Member States on the same document – thus potentially a large number of comments – it uses a tool that allows easily extracting the comments from MS Word documents.

Therefore, for commenting **this** document, please apply the following guidelines:

- All comments are to be written in plain English. Comments provided in other languages cannot, unfortunately, be taken into account.
- The comments must be specific to and must relate to the text (sentence and/or paragraph) being revised.
- Please use simple wording and be as specific, concise and clear as possible in order to avoid ambiguities.
- When referring to specific terms, acronyms, abbreviations that are common in your daily jargon but that are not defined in the *Glossary* document, please define them first.
- Write your comments directly in this MS Word document, by proceeding as follows:
  - First select a word, a part of a sentence or a paragraph (this can be done for example by double-clicking on a word or by dragging your mouse over parts of the text while keeping the left mouse-button pressed).

    Attention:

    Please note that a **minimum of 4 characters** must be selected in order for our commenting tool to grab the comment. Furthermore, comments on diagrams and embedded pictures are also not taken into account. In such cases, please select the caption text underneath the diagram or image.
  - Once a word, part of a sentence or paragraph has been selected, insert an MS Word comment in which you can type your remarks.

    An MS Word comment is typically displayed as a red balloon in the right margin of the document and usually starts with the abbreviation of your name and the timestamp at which the comment is being written. Depending on your version of MS Word, use the following steps for inserting a comment:

    MS Word 2007 and MS Word 2010:

    1. Select the text you would like to comment upon
    2. Open the **Review** ribbon, select **New Comment** in the **Comments** section
    3. In the balloon that appears in the right margin, type your comment
    4. Click anywhere in the document to continue editing the document

    MS Word 2003:

    1. Select the text you would like to comment upon
    2. From the **Insert** menu, select **Comment** (or click on the **New Comment** button on the **Reviewing** toolbar)
    3. In the balloon that appears in the right margin, type your comment
    4. Click anywhere in the document to continue editing the document

    The text will have coloured lines surrounding it, and a dotted coloured line will connect it to the comment. To delete a comment, simply right click on the balloon and select **Delete Comment**.

- Please do not use the MS Word "track changes" tool and do not write your comments as plain text in the MS Word file.
- In case that you want to provide general comments or remarks that are not specific to a part of the text of this document, please provide them into a separate document and/or e-mail.

In case that you need to translate this document to another language, and then translate back your comments to English, please make sure that your comments are provided in the form described above and that they have not been altered or moved to another section of the text during the translation process.

Commenting the technical artefacts

The *WSDL*, *XSD* and *XML* files that are joined to this document can unfortunately not be commented using the same approach as for the MS Word files. Thus, for providing comments to these files, please use the following guidelines:

- General comments affecting **globally** one or more of the technical files: the general comment can be provided in this document in the appropriate sections where the technical files are listed or described.
- Specific comments that affect **specific** elements within the technical files, such as comments on specific *web service* operations, specific *XSD* elements, specific documentation within the technical artefacts, etc. are to be provided in a separate "Comment Inspection Sheet" spread-sheet. An empty MS Excel file is provided together with this document and with the technical artefacts. This file is to be filled in with the following information for each specific comment:
  - Column "#": please indicate here a unique number for your comment
  - Column "File Name": please indicate here the exact name of the WSDL, XSD or XML file to which the comment applies
  - Column "Rvwr": please indicate here the 2-letter code of your Member State
  - Column "Initial Ver.": please indicate here the version of the WSDL, XSD or XML file to which the comment applies (the version is located in the technical files at the top)
  - Column "Original comment & suggestion by reviewer": please indicate here the comment, by first indicating the part of the file concerned (i.e. name of web service operation, name of XSD element, etc.) and then entering the text of the comment

  It is expected that **one comment** is provided **per line** in the spread-sheet. Please avoid typing several comments in the same line, even if they refer to the same technical file or same technical element within the file. This is asked in order to avoid overlooking or forgetting to process comments accidentally.

2 INTRODUCTION

The "Detailed Technical Specifications" are defining the technical ECRIS *service contract*, which consists of a *WSDL* file defining the *web service* operations, several *XSD* files providing a technical implementation of the domain model that has been defined in the "Business Analysis" as well as an XML representation of the common reference tables.

The principles and approaches regarding the design of these technical artefacts follow the ones defined in the "Technical Architecture" document. The XSD type definitions which determine the domain model are split into different namespaces so as to set-up a meaningful separation of concepts. This separation of namespaces using different files is referred to as "packaging" and each pair of "namespace"-"*XSD*" file is named a "package" throughout this document.

The packaging results in the following *XSD* files:

- **"commons-v1.0.xsd":** contains various common definitions of types which are used from other packages
- **"domain-model-v1.0.xsd":** provides the definitions for the main business entities
- **"common-reference-tables-v1.0.xsd":** contains the definitions of the structures required for the *XML* representation of the common reference tables
- **"messages-v1.0.xsd":** provides the definitions of the ECRIS messages that are sent between Member States' central authorities using the *web service* operations
- **"common-reference-tables-v1.0.xml":** provides the *XML* representation of the common reference tables presented in the "Business Analysis" document, with the addition of technical elements (such as fault codes or types of responses) and functional error codes and descriptions

The WSDL file defining the ECRIS web service operations is the following: **"ecris-service-v1.0.wsdl"**

Ensuring a high level interoperability between the platforms of the implementers is a key element of the ECRIS detailed technical specifications. Therefore the *service contract* for ECRIS incorporates **only** those standards that have been defined in the "Technical Architecture" document.

Please note that, as specified in the "Technical Architecture" document, these technical artefacts are thus solely based on the WSDL 1.1 and XSD 1.0 specifications. In particular, specific standards as *RelaxNG, Schematron* and *XSD 2.0* have been discarded for avoiding interoperability issues or implementation constraints. In the same fashion, *web service* extensions such as *WS-Addressing* are also not supported.

Additionally the technical artefacts have been tested using *WSDL*-to-code generators from some of the most popular platforms to ensure compliance. Also the "*WS-I* Basic Profile 1.1" recommendations have been taken under consideration in order to further ensure interoperability.

3 STAKEHOLDERS, ROLES AND RESPONSIBILITIES

3.1 Stakeholders

This section defines the main stakeholders and clarifies their roles and responsibilities towards the detailed technical specifications of ECRIS. The following stakeholders are identified:

- For **the 27 EU Member States**:
  - The Member State delegations which are gathering in the Working Party on Cooperation in Criminal Matters (referred to as "COPEN Working Party")
  - The central authority of every Member State ensuring the implementation of the ECRIS software and its daily operation
  - The designated technical experts which gather regularly in technical workgroups, as described in the "Inception Report" document
- A **central entity** dealing with general support, technical assistance as well as dealing with all horizontal tasks relating to maintenance, evolution and dissemination of information.

3.2 Roles and Responsibilities

3.2.1 Member States

The Member States are engaged in three different ways:

1. The appointed national representatives reconvene in the COPEN Working Group which acts as the appropriate framework structure formally responsible for adoption of **decisions**. It plays the role of "Steering Committee", deciding in particular on:
   - the implementation of appropriate corrective actions in the detailed technical specifications, required for solving problematic situations identified during the development or maintenance of the national ECRIS software
   - the evolution of the detailed technical specifications
2. The central authority of every Member State is in charge of operational tasks, such as the maintenance, development and administration of the national implementation of the ECRIS software. Therefore it has the responsibility to perform the following:
   - providing the updated *service contract* to the central entity, in which the required additional information such as the actual physical address of the service endpoints has been completed
   - ensuring the correct implementation of the *service contract*
   - providing information to the central entity regarding the binary attachments implementation that is currently supported
   - ensuring that the national ECRIS software is available and able to participate appropriately in the message exchanges with other Member States

   Please note that each Member State is free to choose how these tasks are fulfilled. The role does not explicitly imply that a central authority has to perform these tasks either internally or by outsourcing them. This role merely defines the zone of responsibility of the central authority.
3. The appointed national technical experts meet on a regular basis in the technical ECRIS workgroups. They **evaluate** the possibility to apply changes, that are required either for solving problematic situations or for implementing changes requested by the judicial workgroup, to the detailed technical specifications. They act as an **advisory** board on the IT-technical aspects of the ECRIS data exchanges and provide thus support to the decision-making activity of the COPEN Working Party.

   The meetings of the ECRS technical workgroup are chaired by the specific Member State ensuring the Presidency of the Council of the European Union. The appointed technical experts may be assisted by national legal experts and vice versa since technical and legal issues are not always separate from each other.

It is recommended that the ECRIS technical workgroups meet **at least twice a year**, following the same frequency and combined with the ECRIS judicial workgroups. However these workgroups need to adopt a large degree of flexibility and may reconvene as often as required - especially if urgent matters arise.

3.2.2 Central Entity

In accordance with the decentralised nature of the ECRIS data exchanges, with the findings of the preliminary analyses and other deliverables of the *ECRIS Technical Specifications* project, a central entity is deemed necessary for effectively coordinating the works, providing organisational support, performing some technical maintenance and updates of the detailed specifications and disseminating information. The central entity can only perform changes to the detailed technical specifications if it is explicitly mandated by the COPEN Working Party to do so.

This central entity is responsible for the following activities:

- organising and coordinating the technical workgroup meetings and workshops; agenda items of these workgroups can be proposed by participating Member States; to that end, the central entity notifies sufficiently early the Member States which can have the opportunity to propose agenda items in a timely manner
- implementing the changes decided in the COPEN Working Group meetings in those technical artefacts which have been handed over to the central entity by the COPEN, and transmitting them to the national technical experts for revision
- under the mandate of the COPEN, keeping the documentation of the *ECRIS Technical Specifications* up-to-date and especially updating them in accordance with the decisions taken in the COPEN Working Group meetings
- circulating the updated versions of the technical artefacts which have been handed over to the central entity by the COPEN and related documentation to the appointed experts of the Member States
- receiving, validating and publishing the national versions of the *service contract,* the HTTPS security certificates as well as information regarding the implementation of binary exchanges and versions supported by the central authorities of the Member States

Please note that in regards to circulation of information, the central entity is free to choose the means considered most appropriate for the task, keeping though always in mind that these technical means must not create technical dependencies with the national ECRIS software implementations. In particular, central repositories such as UDDI registries have been considered and already discarded in the "Technical Architecture" document due to the fact that they would create a single point of failure for the automated ECRIS data exchanges.

According to article 3(7) of the Council Decision 2009/316/JHA, the **European Commission** "… shall provide general support and technical assistance, including the collection and drawing up of statistics …" and shall thus act as central entity for the tasks listed above that are considered as being part of general support and technical assistance. In addition, the European Commission has the mandate of producing the *ECRIS Reference Implementation* and of operating and maintaining the sTESTA network. For the other tasks to be carried out at a central level but that do not fall into the responsibility of the European Commission, the COPEN Working Party needs to mandate the appropriate body.

❗	Please note that establishing the detailed description of the mission of the central entity is essential for a successful implementation and functioning of ECRIS but is out of scope of the *ECRIS Technical Specifications* project and is not included in this document.

4 DETAILED TECHNICAL SPECIFICATIONS

The following sections provide an in-depth presentation of the practices, approaches and solutions that are used in the technical artefacts so as to ensure interoperability and reduce information redundancy. Samples and diagrams are provided where necessary so as to illustrate the concepts being described.

4.1 General Information

The following sub-chapters provide general information about the technical artefacts that constitute the detailed technical specifications of ECRIS.

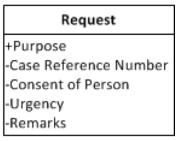
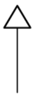
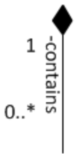
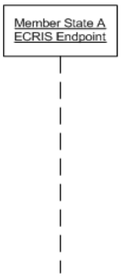
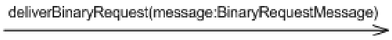
4.1.1 UML Diagrams

The diagrams in the next chapters use the UML formalism but have been simplified in order to facilitate the reading. In particular, the properties of the entities are not always shown in all diagrams.

Two types of diagrams are used:

- The class diagrams illustrate the definitions provided in the technical artefacts in terms of information entities and messages
- The sequence diagrams illustrate the kinematics (i.e. sequence of operations) to be performed, using the web service operations defined in the *WSDL* file

The following legend applies to these diagrams:

SYMBOL	DESCRIPTION
 Request +Purpose -Case Reference Number -Consent of Person -Urgency -Remarks	This symbol represents an information entity. Within the box, the simple properties contained in the entity are listed. The "+" sign indicates a mandatory property while the "-" sign indicates an optional property. For entities that are derived from their parent entity by extension rather than by restriction, the properties of the parent are not repeated in the definition.
	This type of arrow indicates that an entity extends or specialises another entity. The arrow head points to the general or abstract entity that is being extended. When used for entities that derive from a parent entity by restriction, an additional caption "restricts" is added for clearly marking the difference.
	This type of arrow indicates that an entity is composed of/contains another entity. The arrow head points to the entity being the container of the entity at the other end.
	This symbol represents an instance of an object. In the context of this document, the name refers to an instance of an ECRIS web service endpoint as a whole.
	This arrow represents an asynchronous method call with no return value. The name of the operation is stated on top of

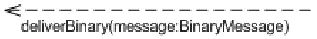
	the arrow, with the name and type of the parameter being mentioned between parentheses.
 deliverBinary(message:BinaryMessage)	This symbol represents an asynchronous return operation. The name of the operation is stated below the arrow, with the name and type of the parameter being mentioned between parentheses.

Table 1 – Legend for UML class diagrams

4.1.2 «Dummy» Values for Mandatory Elements

As defined in the "Business Analysis" document, each mandatory element being defined in the XSD files needs to support a "dummy" value which carries the meaning that the information is unavailable to the sender of the message.

For each such element, the XSD file defines an appropriate default value.

By convention, the following values are systematically used as defaults, depending on the technical type of the element:

- Value "UNKNOWN" for textual elements based on `xsd:string` (such as `commons:StringType`, `commons:MultilingualTextType` or `dm:YesNoUnknownStringEnumerationType`)
- Value "1800-01-01" for date elements based on `xsd:date` (i.e. `commons:StrictDateType`)

The following specific situations are however to be noted:

- The value "application/nist" has been set as default value for the MIME type of binary attachments (i.e. element `msg:BinaryAttachmentType:BinaryAttachmentMimeType`)
- The value "0" (i.e. means "unknown") has been defined for the `dm:PersonType:PersonSex` element
- For data elements defining a reference to a value contained in common reference table (such as `dm:PlaceType:PlaceCountryReference`), the value has been set to the technical identifier of the entry in the corresponding table that has the meaning "unknown" (example: value "CO-00-999-XXX" for a reference to a country)

Please note also that for very specific mandatory fields such default values are not defined purposefully. Indeed, such fields must be filled with a proper value otherwise the ECRIS data exchanges cannot take place from a technical standpoint. This concerns mainly the technical message meta-data fields such as message ID's, timestamps, correlation information, etc.

4.1.3 Transactions

The term "transaction" used throughout this document is to be understood as a sequence of individual ECRIS messages, functionally interrelated into a dialogue between two Member States. Such a dialogue is always characterised by a starting and an ending point. In particular, the word "transaction" in this document is not to be confused with "database transactions" where all actions are technically related and undone in case of an error in the execution.

The following sequence of messages illustrates a transaction as understood in this document:

The **starting point** consists of a request message sent by Member State "A" to Member State "B". "B" sends back the request deadline and then sends a "request additional ID info" message. "A" replies to this by sending back the requested identification information through an "additional ID info" message. "B" then sends the request response to "A". Because of an error in the request response, "B" cancels the previously transmitted request response message and sends a new one using corrected information. This latter response request message is the **ending point** of this transaction.

An ECRIS transaction thus **starts** when any of the following actions are occurring:

- a valid request message is received
- a valid notification message is received

According to the definitions provided in this document, including the cancellation mechanism defined later, an ECRIS transaction is to be considered as **finished** for both sending and receiving Member States when one of the following events are occurring:

- the legal deadline for the request response has passed
- the operational deadline for the notification response has passed
- a valid cancellation message is received as a response to a previously sent request
- a valid cancellation message is received as a response to previously sent notification
- a valid functional error is received as a response to a previously sent request
- a valid functional error is received as a response to a previously sent notification

4.2 Packages and Type Definitions

To avoid information redundancy and allow flexibility and extensibility, abstractions are used as much as possible in the design of the domain model. The principles of encapsulation, inheritance and polymorphism, as described in the "Technical Architecture" document, are applied to the maximum possible extent.

The following chapters provide an explanation on how the aforementioned principles are applied. The separation of the chapters is based on the packaging.

4.2.1 Commons

A common problem when creating type definitions and separating them into packages is that the dependencies created between those packages can result in circular dependencies. For example, if type definition "AType" in package "A" depends on "BType" of package "B", the dependency is considered one-way, since package "B" does not depend on package "A". If now a new type definition "CType" is added in package "B", which depends on "AType", a circular dependency between the two packages is created. Generally speaking, such circular dependencies are considered an anti-pattern as they introduce various weaknesses to the model. Concretely, such interdependencies create a tight coupling that reduces or makes impossible re-using one of the packages without the other. Furthermore, this kind of coupling can introduce a "domino" effect, where a small change in one package can cause major reworking in the depending packages.

In order to avoid such circular dependencies, it is common practice to introduce a separate package which contains definitions that are used in more than one other package. In the detailed technical specifications, this package is named "commons" and the definitions created therein are placed in the namespace "**`http://ec.europa.eu/ECRIS/commons-v1.0`**".
Additionally, with a careful selection of the type definitions included in the "commons" package, the "domino" effect mentioned above as a negative side-effect is turned into a positive feature. This is achieved by placing in the "commons" package proxies for the *XSD* base simple types, such as `xsd:date` or `xsd:string`, which are used throughout the packages. In doing so, if a structural change is required in such a base type there is no need to duplicate the change to all packages using it.

Please note that for future evolutions, the rule that should be followed in order to keep the "commons" package consistent is the following:

- Before adding a new type in the "commons" package, the following two checks must be performed and have a positive outcome
  - The type is used by at least two packages.
  - The type does not represent a specific business concept. Indeed, if a business-related type has to be re-used, the implementers should rather consider defining a new, separate package which represents the business domain this type belongs to.

The following UML class diagram provides an overview of **all** the types that are currently defined in the "commons" package:

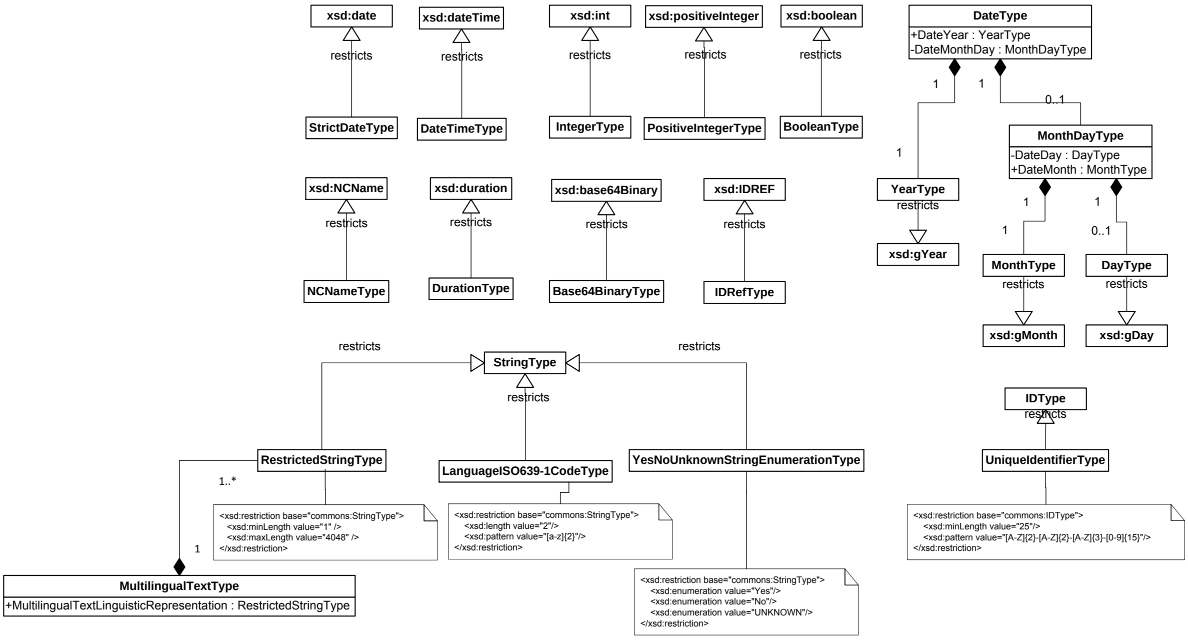


Figure 1 – Types defined in the "commons" package

Please note that most of the types contained in this package are actually "proxy" types deriving directly from native XSD types. It is to be noted though that a specific "DateType" type has been defined so as to allow defining dates that are only partially complete (i.e. year and month without day), whereas the "StrictDateType" only allows complete dates to be provided. The "DateType" type is typically used for dates that are defined in the "Business Analysis" as allowing incomplete values, such as for instance the start date of an offence. In addition, the free-text elements have been limited to 4048 characters in length by defining the "RestrictedStringType" element which is used as basis for the "MultilingualTextType" element.

4.2.2 Common Reference Tables

The "common reference tables" package contains the structural **definition** of **all** the common reference tables used in ECRIS. The **content** of these reference tables is provided separately in one single *XML* file. This separation between structural definition and content has been chosen so as to allow easily changing the content of the common reference tables without need to modify the XSD definition of it. It is indeed expected that the XML file will be regularly modified in the scope of the regular maintenance of the ECRIS software.

The namespace defined for the "common reference table" package is "**`http://ec.europa.eu/ECRIS/common-reference-tables-v1.0`**".

Usage of `xsd:ID` and `xsd:IDREF`

One of the main difficulties when designing the *XSD* structure supporting the *XML* representation of the common reference tables is to keep on one hand the main principles of reusability and extensibility while on the other hand ensuring **at design time**, **by the structure** that the current but also future versions of common reference tables contain valid information on a technical level. Concretely, this mechanism ensures that the technical identifiers used within those tables are unique and when used for cross-referencing (such as cities referencing countries) the linkage between such entities is also using valid identifiers. Ensuring the uniqueness and validity of such technical identifiers indeed ensures on one hand the validity of the common reference values but also avoids data corruption when referencing to these reference values in other entities of the domain model.

The mechanism described above is implemented in the XSD definitions as follows:

- The technical identifiers of the entities defined in the *XSD* schema extend the `commons:IDType` which extends the `xsd:ID` type. This ensures the uniqueness of technical identifiers when used within the same *XML* document.

  As a consequence, if an implementer adds values to one of the common reference tables with an *XML* authoring tool that performs validation, this authoring tool will notify immediately of the error if the implementer tries to reuse an already defined technical identifier in the XML by mistake.
- In order to ensure that referencing between two entities is done with an existing technical identifier, an additional reference type is created by extending the `commons:IDRefType` which in turns extends the `xsd:IDREF` type.

  In the same way, if an implementer adds values to one of the common reference tables with an *XML* authoring tool that performs validation, this authoring tool will also notify immediately of the error if the implementer tries to establish by mistake a relation between an entity with another entity of the wrong type or using a wrong reference (for example when referencing a country in the common reference table listing cities).

Please note at this stage that such validation **at design time** can **only** work when using an appropriate *XML* authoring tool that properly implements the derived data types as specified in "XML Schema Part 2: Datatypes – Second Edition" of the *XSD* 1.0 specification, and in particular the "xsd:ID" and "xsd:IDREF" types. A few examples of such *XML* authoring tools are Altova XMLSpy, XMLNotepad and Oxygen XML.

In addition, please note that this mechanism of validation of proper and unique technical references yields correct results **only if** both the technical identifier and the reference are declared in the same *XML* document. Unfortunately this is due to the fact that the *XML* specification does not provide a way to import external documents when authoring or when validating an *XML* document. Thus, if the technical identifier used by a reference is not available to the parser within the same *XML* document, the document will be considered invalid by this parser. For this specific reason the content of all common reference tables are provided in a **single *XML* document**.

Furthermore, the approach followed in order to create the technical identifiers used in the common reference tables has to ensure that on one hand each identifier is unique and on the other hand has to facilitate tracking changes in existing values. Taking that under consideration, the general logic that is used for creating technical identifiers is the following:

[A-Z]{1,3}-[0-9]{2}-*

where

- **[A-Z]{1,3}** is a string value of 1 to 3 characters derived from the name of the specific entity for which the technical identifier is created for
- **[0-9]{2}** is a two digit number that should be incremented by 1 when a new version of an existing value is added as a new entry in the reference table
- ***** are various other entity specific elements that can ensure the uniqueness of the technical identifier

As a concrete example, the technical identifier for offence categories uses the following restriction

O-[0-9]{2}-[0-9]{6}

where

- **O** is the first letter of the word "**O**ffence"
- **[0-9]{2}** is the aforementioned versioning number. For example, the categories of offences currently defined in the common reference table all have this part set to "00". If in the future one of the existing categories needs to be redefined with a slightly different judicial meaning, a new entry can be added with the same category code but incrementing this part of the identifier to "01". This allows having a kind of logical versioning when defining new entries in the common reference tables.

- **[0-9]{6}** is a 6 digit number depicting the offence category code as it is defined in Annex A of the Council Decision 2009/316/JHA

Usage of an external reference type extending `xsd:string`

The main role of the common reference tables is to provide a standardised and codified representation of all values commonly understood in the same way by the Member States and to be used in the ECRIS data exchanges.

Performing this task requires using the technical identifiers defined for the records in the common reference tables within the body of the *XML* messages that are sent through the *web service* operations. However reusing the reference types (extending `xsd:IDREF`) or the technical identifiers themselves (extending `xsd:ID`) does not work for the same reasons as mentioned above. Indeed, using an extension of `xsd:IDREF` in the *XML* body of a message would require that the technical identifier being referenced is also declared in the same *XML* document. This would imply duplicating the definition of the common reference value being used inside the same *XML* document. Using an extension of `xsd:ID` in the *XML* body of a message would also not work since a value of that type can occur only once within an XML document.

The approach that has been chosen is to define specific element types used for the referencing of the common reference values in the *XML* body of the ECRIS messages, based on the `commons:StringType` element which is a simple text element on which using simple text elements (of types extending `commons:StringType`) on which the same restrictions of format and content are applied than for the technical identifier itself.

While this solution solves the referencing of common reference values in the *XML* body of messages without requiring duplicating the content of the common reference tables into each message, it also has the following drawbacks:

- When validating the XML message, the parser will not be able to automatically verify that a value used as a reference indeed matches an existing technical identifier in the common reference tables. Verifying that a reference points to a proper record in the expected common reference table needs thus to be done programmatically in a second step during the functional validation of the message.
- In addition, the same definition of restriction of format needs to be included up to 3 times in the XSD definition of the types: once for defining the technical identifier itself (based on `commons:IDType` → `xsd:ID`), once for defining the internal reference to the technical identifier (based on `commons:IDRefType` → `xsd:IDREF`) and once more for defining the external reference to the technical identifier (based on `commons:StringType` → `xsd:string`).

The following UML class diagram illustrates the definition of the two common reference tables' entities "CityType" and "CountryType":

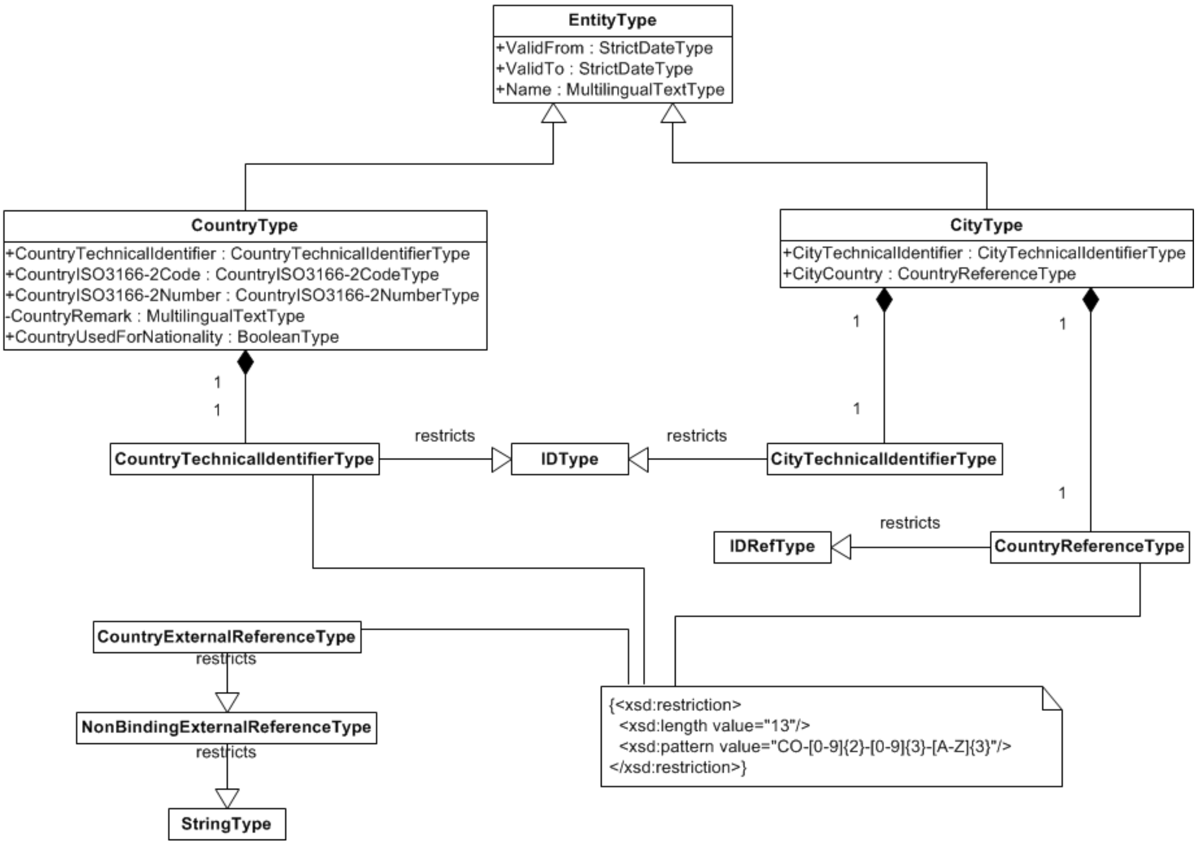

EntityType
+ValidFrom : StrictDateType
+ValidTo : StrictDateType
+Name : MultilingualTextType
CountryType
+CountryTechnicalIdentifier : CountryTechnicalIdentifierType
+CountryISO3166-2Code : CountryISO3166-2CodeType
+CountryISO3166-2Number : CountryISO3166-2NumberType
-CountryRemark : MultilingualTextType
+CountryUsedForNationality : BooleanType
CityType
+CityTechnicalIdentifier : CityTechnicalIdentifierType
+CityCountry : CountryReferenceType
1
1
1
1
1
1
CountryTechnicalIdentifierType
restricts
IDType
restricts
CityTechnicalIdentifierType
IDRefType
restricts
CountryReferenceType
CountryExternalReferenceType
restricts
NonBindingExternalReferenceType
restricts
StringType
{<xsd:restriction>
<xsd:length value="13"/>
<xsd:pattern value="CO-[0-9]{2}-[0-9]{3}-[A-Z]{3}"/>
</xsd:restriction>}

Figure 2 – Definition of "CountryType" and "CityType" common reference entities

List of common reference tables provided

Concretely, the following definitions for common reference tables are provided:

- **CentralAuthorities:** table defining the official central authorities of the Member States
- **Languages:** table defining the official languages of the Member States
- **Countries:** table defining **all** countries based on the ISO 3166-1 standard (using the 2-letter code and 3-digit numeric representation)

  (see http://en.wikipedia.org/wiki/ISO_3166-1)
- **Currencies:** table defining **all** national currencies based on the ISO 4217 standard

  (see http://en.wikipedia.org/wiki/ISO_4217)
- **CountrySubdivisions:** table defining the country subdivisions (i.e. provinces, counties, districts, etc.) of the Member States, based on the ISO 3166-2 standard

  (see http://en.wikipedia.org/wiki/ISO_3166-2)
- **Cities:** table defining the biggest cities in the 27 Member States

  Please note that this table should be expanded over time so as to add the city names most frequently used in the ECRIS data exchanges.
- **OffenceCategories:** table providing the categories of offences as defined in Annex A of the Council Decision 2009/316/JHA
- **OffenceLevelsOfParticipation:** table providing the categories for the level of participation in the offence as defined in the parameters in Annex A of the Council Decision 2009/316/JHA
- **OffenceLevelsOfCompletion:** table providing the categories for the level of completion of the offence as defined in the parameters in Annex A of the Council Decision 2009/316/JHA
- **SanctionCategories:** table providing the categories of the sanctions as defined in Annex B of the Council Decision 2009/316/JHA

- **SanctionTypesOfSuspension:** table providing the categories for the type of suspension of the sanction as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **SanctionNatures:** table providing the categories for the type of the sanction (i.e. penalty or measure) as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **SanctionAlternativeTypes:** table providing the types of alternative sanctions as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **RequestPurposeCategories:** table providing the common categories of purposes of requests as defined in the "Business Analysis" document
- **IdentificationDocumentCategories:** table providing the common categories of identification documents used in the Member States, as defined in the "Business Analysis" document
- **DecisionChangeTypes:** table providing the types of changes that can affect a conviction as a whole or one of the sanctions within the conviction, as defined by the parameters in Annex B of the Council Decision 2009/316/JHA.
- **RequestResponses:** table providing the possible types of responses that can be sent for requests

  Please note that it contains values for positive (i.e. the request could be processed) and negative responses (i.e. the request denial and request problem messages defined in the "Business Analysis" document).
- **NotificationResponses:** table providing the possible types of responses that can be sent for notifications

  Please note that it contains values for positive (i.e. the notification receipt) and negative responses (i.e. the notification problem message defined in the "Business Analysis" document).
- **RequestAdditionalInfoResponses:** table providing the possible types of responses that can be sent for the request for additional identification information

  Please note that it contains values for positive (i.e. the additional info available) and negative responses (i.e. the additional info is not available).
- **FunctionalErrors:** table providing the list of functional error codes that can possibly occur during the functional validation of the ECRIS messages
- **Faults:** table providing the list of the supported SOAP fault codes with their descriptions
- **SanctionToSanctionsRelationshipTypes:** table providing the list of the supported relationship types for the sanction-to-sanction relations (i.e. the "SanctionToSanctionsRelationshipType" definition)
- **RequestingAuthorityTypes:** table providing the list of supported types for requesting authorities

4.2.3 Domain Model

Most of the structured information entities defined in the "Business Analysis" document are contained in the "domain model" package, with the exception of the messages and entities related conceptually to other packages such as the "Contact Person" (which is contained in the "messages" package).

The namespace defined for the "domain model" package is "**`http://ec.europa.eu/ECRIS/domain-model-v1.0`**".

The table listing the business entities defined in the "Business Analysis" and their technical counterparts is provided below. Where appropriate, the technical counterpart is prefixed with the namespace of the package in which it is defined.

Business entity	Technical counterpart
Person	PersonType

Contact Person	{http://ec.europa.eu/ECRIS/messages-v1.0}ContactPersonType
Requesting Authority	{http://ec.europa.eu/ECRIS/messages-v1.0}RequestingAuthorityType
Place	PlaceType
Address	PersonAddressType
Identification Document	IdentificationDocumentType
Alias	PersonAliasType
Conviction	ConvictionType
Decision	DecisionType
Offence	OffenceType
Sanction	SanctionType
Suspension	SanctionSuspensionType
Interruption	SanctionInterruptionType

Table 2 – Business entities and their technical counterparts

This package defines the entities by following closely the principles of extensibility and re-use of defined structures. In particular, abstract types are defined for re-using common definitions.

The decisions, offences and sanctions are structurally defined within the "ConvictionType" element. In addition, please note that the various relations between sanctions, offences and decisions are also defined within the "ConvictionType" entity, using the optional element "ConvictionRelationship" of type "AbstractRelationshipType". This approach ensures that all currently required types of relationships are supported and that adding new types of relationships in the future remains easy to implement. The concrete types of relations that can be established between the main entities above and which derive from the "AbstractRelationshipType" are the following:

- "DecisionToOffencesRelationshipType" allows establishing a relation between a decision and one or more offences
- "DecisionToSanctionsRelationshipType" allows establishing a relation between a decision and one or more sanctions
- "SanctionToOffencesRelationshipType" allows establishing a relation between a sanction and one or more offences
- "SanctionToSanctionsRelationshipType" allows establishing a relation between a sanction and one or more sanctions. Please note that this specific relation features a mandatory attribute "relationshipType" that indicates the type of the relationship by linking to an entry in the common reference table "SanctionToSanctionsRelationshipTypes"

Technical identifiers of main entities

In the implementation of the types above, it is important to notice that the elements "DecisionType", "OffenceType" and "SanctionType" contain each a technical identifier to be used for cross-referencing within the same *XML* document. Such technical identifiers **must be unique within an XML document**, but not necessarily across several ECRIS messages.

The "ConvictionType" carries a technical identifier that **must be unique across all ECRIS messages** and that can be used for relating to convictions externally in other *XML* documents.

As for the common reference tables, a technical identifier based on `commons:IDType` → `xsd:ID` is defined for the aforementioned entities. In addition, an internal reference to the technical identifier based on `commons:IDRefType` → `xsd:IDREF` is defined for "OffenceType" and "SanctionType" whereas the `crt:NonBindingExternalReferenceType` → `commons:StringType` is used for referencing entities of type "ConvictionType.

These identifiers have been defined with the following formats:

- For convictions: [A-Z]{2}-C-[0-9]{15} where the two first letters represent the country code of the Member State issuing the conviction. For example, if Greece is issuing a conviction, the identifier would look like the following:

  GR-C-554672903178462
- For offences: O-[0-9]{5} (example: O-80033)
- For sanctions: S-[0-9]{5} (example: S-71596)
- For decisions: D-[0-9]{5} (example: D-32157)

(where [0-9]{5} is a 5-digit number ranging from 00000 to 99999)

Please note also that no external reference type is defined for "OffenceType", "SanctionType" and "DecisionType" entities; indeed this is not necessary since these are always declared in the same *XML* document in an ECRIS message.

Please note that the fingerprints information has been defined on the message level in the XSD definition rather than in the "Person" entity so as to be able to properly reuse the structure.

Illustration

The following UML class diagram provides an overview of **some** of the entities defined the domain model, around the "ConvictionType" definition, so as to illustrate the general principle followed in the domain model definition:

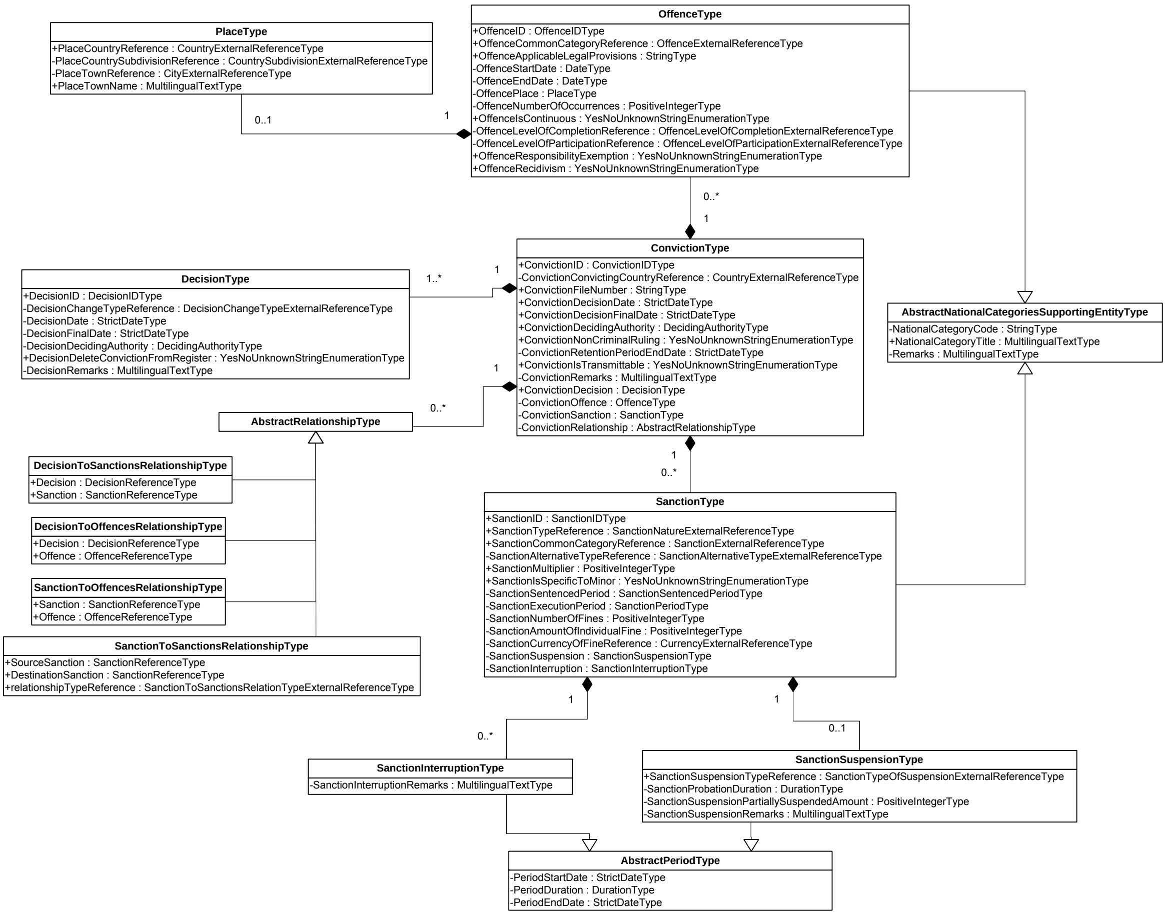

PlaceType
+PlaceCountryReference : CountryExternalReferenceType
-PlaceCountrySubdivisionReference : CountrySubdivisionExternalReferenceType
-PlaceTownReference : CityExternalReferenceType
+PlaceTownName : MultilingualTextType
0..1
1
OffenceType
+OffenceID : OffenceIDType
+OffenceCommonCategoryReference : OffenceExternalReferenceType
+OffenceApplicableLegalProvisions : StringType
-OffenceStartDate : DateType
-OffenceEndDate : DateType
-OffencePlace : PlaceType
-OffenceNumberOfOccurrences : PositiveIntegerType
+OffenceIsContinuous : YesNoUnknownStringEnumerationType
-OffenceLevelOfCompletionReference : OffenceLevelOfCompletionExternalReferenceType
-OffenceLevelOfParticipationReference : OffenceLevelOfParticipationExternalReferenceType
+OffenceResponsibilityExemption : YesNoUnknownStringEnumerationType
+OffenceRecidivism : YesNoUnknownStringEnumerationType
0..*
1
DecisionType
+DecisionID : DecisionIDType
-DecisionChangeTypeReference : DecisionChangeTypeExternalReferenceType
-DecisionDate : StrictDateType
-DecisionFinalDate : StrictDateType
-DecisionDecidingAuthority : DecidingAuthorityType
+DecisionDeleteConvictionFromRegister : YesNoUnknownStringEnumerationType
-DecisionRemarks : MultilingualTextType
1..*
1
ConvictionType
+ConvictionID : ConvictionIDType
-ConvictionConvictingCountryReference : CountryExternalReferenceType
+ConvictionFileNumber : StringType
+ConvictionDecisionDate : StrictDateType
+ConvictionDecisionFinalDate : StrictDateType
+ConvictionDecidingAuthority : DecidingAuthorityType
+ConvictionNonCriminalRuling : YesNoUnknownStringEnumerationType
-ConvictionRetentionPeriodEndDate : StrictDateType
+ConvictionIsTransmittable : YesNoUnknownStringEnumerationType
-ConvictionRemarks : MultilingualTextType
+ConvictionDecision : DecisionType
-ConvictionOffence : OffenceType
-ConvictionSanction : SanctionType
-ConvictionRelationship : AbstractRelationshipType
AbstractNationalCategoriesSupportingEntityType
-NationalCategoryCode : StringType
+NationalCategoryTitle : MultilingualTextType
-Remarks : MultilingualTextType
1
0..*
AbstractRelationshipType
DecisionToSanctionsRelationshipType
+Decision : DecisionReferenceType
+Sanction : SanctionReferenceType
DecisionToOffencesRelationshipType
+Decision : DecisionReferenceType
+Offence : OffenceReferenceType
SanctionToOffencesRelationshipType
+Sanction : SanctionReferenceType
+Offence : OffenceReferenceType
SanctionToSanctionsRelationshipType
+SourceSanction : SanctionReferenceType
+DestinationSanction : SanctionReferenceType
+relationshipTypeReference : SanctionToSanctionsRelationTypeExternalReferenceType
1
0..*
SanctionType
+SanctionID : SanctionIDType
+SanctionTypeReference : SanctionNatureExternalReferenceType
+SanctionCommonCategoryReference : SanctionExternalReferenceType
-SanctionAlternativeTypeReference : SanctionAlternativeTypeExternalReferenceType
+SanctionMultiplier : PositiveIntegerType
+SanctionIsSpecificToMinor : YesNoUnknownStringEnumerationType
-SanctionSentencedPeriod : SanctionSentencedPeriodType
-SanctionExecutionPeriod : SanctionPeriodType
-SanctionNumberOfFines : PositiveIntegerType
-SanctionAmountOfIndividualFine : PositiveIntegerType
-SanctionCurrencyOfFineReference : CurrencyExternalReferenceType
-SanctionSuspension : SanctionSuspensionType
-SanctionInterruption : SanctionInterruptionType
1
1
0..*
0..1
SanctionInterruptionType
-SanctionInterruptionRemarks : MultilingualTextType
SanctionSuspensionType
+SanctionSuspensionTypeReference : SanctionTypeOfSuspensionExternalReferenceType
-SanctionProbationDuration : DurationType
-SanctionSuspensionPartiallySuspendedAmount : PositiveIntegerType
-SanctionSuspensionRemarks : MultilingualTextType
AbstractPeriodType
-PeriodStartDate : StrictDateType
-PeriodDuration : DurationType
-PeriodEndDate : StrictDateType

Figure 3 – Domain Model – "ConvictionType" and related types

4.2.4 Messages

All messages that can be sent from one Member State to another one in the *web service* operations have been defined in the "messages" package and are placed in the namespace: **"http://ec.europa.eu/ECRIS/messages-v1.0"**.

Here also the principles of abstraction and re-use are applied: a base abstract type named "AbstractMessageType" defines the basic message structure composed of message meta-data and message data. Please note that the abstract message data type is defined as being of type "xsd:anyType". This is done for allowing defining more specific message data types by restriction. The message meta-data property is however strongly typed.

The following UML class diagram illustrates the composition and structure of the "AbstractMessageType":

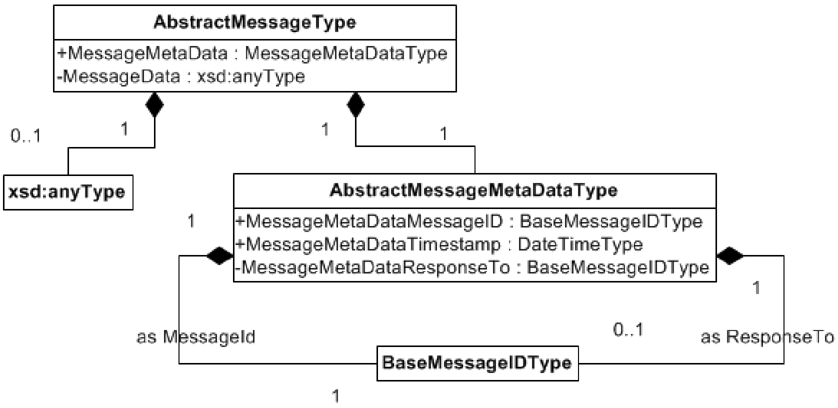

AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
0..1
1
1
1
xsd:anyType
AbstractMessageMetaDataType
+MessageMetaDataMessageID : BaseMessageIDType
+MessageMetaDataTimestamp : DateTimeType
-MessageMetaDataResponseTo : BaseMessageIDType
1
1
as MessageId
0..1
as ResponseTo
BaseMessageIDType
1

Figure 4 – Messages – "AbstractMessageType"

Following this approach, each specific message derived from this type **by restriction** must define its message data structure, if any, as well as an extension of the message meta-data type.

The following UML diagrams illustrate the definitions of the concrete messages and how they are derived from "AbstractMessageType":

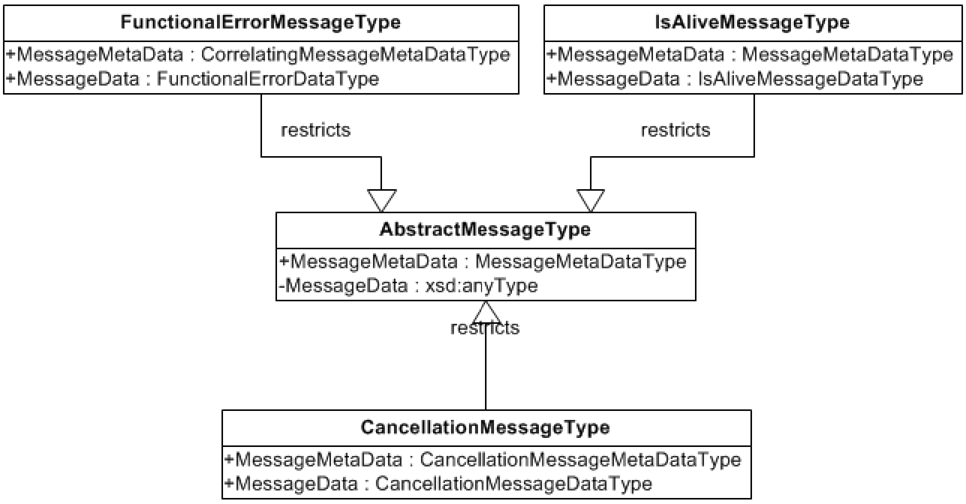

FunctionalErrorMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : FunctionalErrorDataType
IsAliveMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : IsAliveMessageDataType
restricts
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
restricts
CancellationMessageType
+MessageMetaData : CancellationMessageMetaDataType
+MessageData : CancellationMessageDataType

Figure 5 – Messages – Concrete message types for "IsAlive", "FunctionalError" and "Cancellation" implementation messages

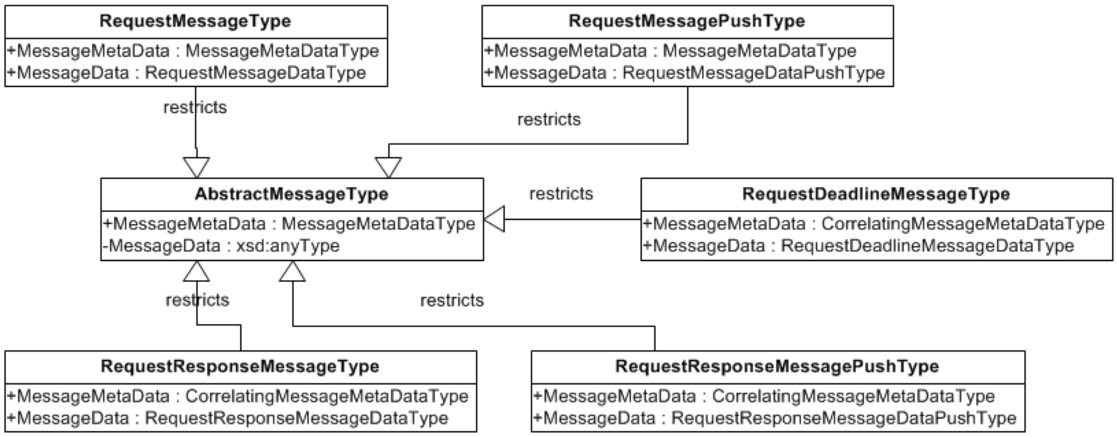

RequestMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : RequestMessageDataType
RequestMessagePushType
+MessageMetaData : MessageMetaDataType
+MessageData : RequestMessageDataPushType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
RequestDeadlineMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestDeadlineMessageDataType
RequestResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestResponseMessageDataType
RequestResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestResponseMessageDataPushType

Figure 6 – Messages – Concrete message types for the "request" business process

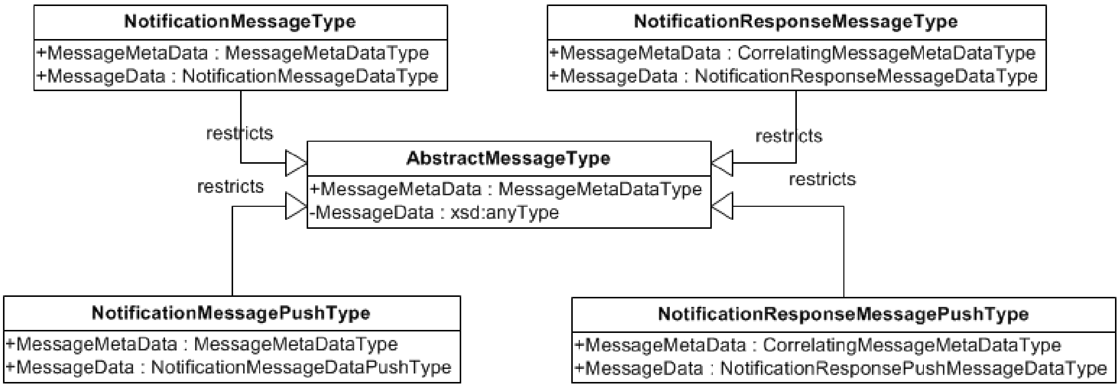

NotificationMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : NotificationMessageDataType
NotificationResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : NotificationResponseMessageDataType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
NotificationMessagePushType
+MessageMetaData : MessageMetaDataType
+MessageData : NotificationMessageDataPushType
NotificationResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : NotificationResponsePushMessageDataType

Figure 7 – Messages – Concrete message types for the "notification" business process

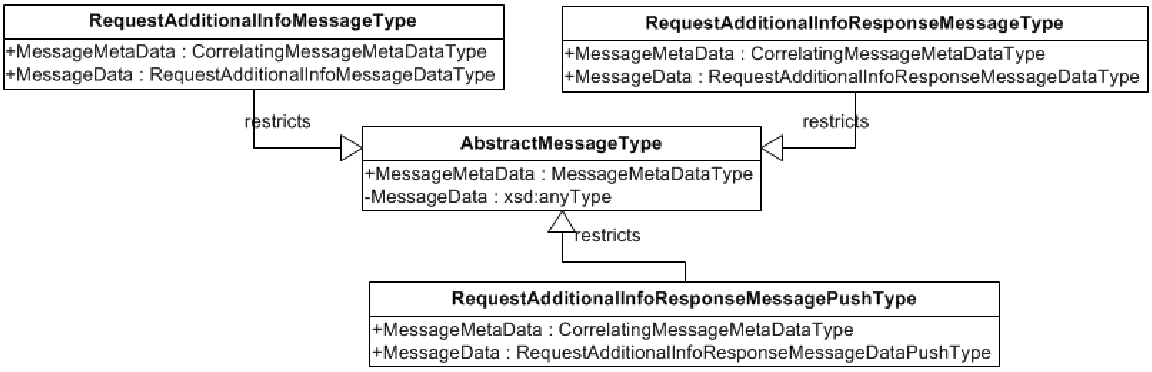

RequestAdditionalInfoMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoMessageDataType
RequestAdditionalInfoResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoResponseMessageDataType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
RequestAdditionalInfoResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoResponseMessageDataPushType

Figure 8 – Messages – Concrete messages of the "search person" sub-process

In the diagrams above, please note that each message defines its own message data type, according to the information structure required specifically by the message.

Business messages versus technical messages

As mentioned in the "Business Analysis", the business messages do not necessarily have a one to one correspondence with the messages defined in the technical *service contract.* Indeed, messages that carry identification information have to support binary attachments for fingerprints and thus require additional messages so as to implement both "Push" and "Pull" approaches as defined in the "Technical Architecture" document. Additionally, with the interest of keeping the number of message types and operations to a minimum and avoid information or structural redundancy, several operations and messages described in the "Business Analysis" are implemented using one message and operation definition.

The following table provides the messages defined in the "Business Analysis" and their technical counterparts:

Business Message	Technical Message
Notification	NotificationMessageType NotificationMessagePushType
Notification Problem	NotificationResponseMessageType
Notification Receipt	NotificationResponseMessageType NotificationResponseMessagePushType
Request	RequestMessageType RequestMessagePushType
Request Deadline	RequestDeadlineMessageType
Request Denial	RequestResponseMessageType
Request Problem	RequestResponseMessageType
Request Response	RequestResponseMessageType RequestResponseMessagePushType
Request Additional ID Info	RequestAdditionalInfoMessageType
Additional ID Info	RequestAdditionalInfoResponseMessageType RequestAdditionalInfoResponseMessagePushType
Additional ID Info Unavailable	RequestAdditionalInfoResponseMessageType

Table 3 – Business messages and their technical counterparts

The various possible responses to a notification, to a request and to a request for additional identification information have been grouped into the technical messages "NotificationResponse", "RequestResponse" and "RequestAdditionalInfoResponse". In each such technical response message, a response type indicates the nature of the response and provides the more fine-grained result depending on the type of message. The definition of the response types for each message is provided in the following common reference tables which can be expanded easily for example for adding specific causes for not providing a positive response:

- RequestResponses
- NotificationResponses
- RequestAdditionalInfoResponses

Grouping the response messages can prove useful from the point of view of implementers, especially when monitoring the transactions. Indeed, the "notification response" and "request response" messages are always closing the ongoing "notification" or "request" transaction.

For each message carrying personal identification information, a corresponding "push" message is defined. The simple message may carry only meta-data about the attached binary fingerprints file while the "push" message may carry the actual binary file. The principle is further detailed later in this document.

Message Meta-Data

As defined in the "Technical Architecture" document, the functional information is to be clearly separated from the technical information in the body of the *XML* message. To that end, a series of data structures are defined for carrying the technical meta-data of the message.

These meta-data structures are based on the "AbstractMessageMetaDataType" described above and contain the following properties:

- "MessageMetaDataMessageID" carries the technical identifier of the message. Please note that the ECRIS messages exchanged **are required** to be **uniquely identifiable** not only on a national level, but across all ECRIS message exchanges so as to be able to properly correlate them.

The technical identifier type is always mandatory. The "BaseMessageIDType", from which all other ID types are derived, defines the following format:
[A-Z]{2}-[A-Z]{2}-[A-Z]{3}-[0-9]{15} where:

- **[A-Z]{2}** is the two letter code of the sending Member State in **capitals**
- **[A-Z]{2}** is the two letter code of the receiving Member State in **capitals**
- **[A-Z]{3}** is a 3 character code for the message type. The following values are valid
  - REQ for requests

    (types "RequestMessageType", "RequestPushMessageType")
  - RDL for request deadlines

    (type "RequestDeadlineMessageType")
  - RRS for request responses

    (types "RequestResponseMessageType", "RequestResponseMessagePushType")
  - NOT for notifications

    (types "NotificationMessageType", "NotificationMessagePushType")
  - NRS for notification responses

    (types "NotificationResponseMessageType", "NotificationResponsePushMessageType")
  - RAI for requests for additional information

    (type "RequestAdditionalInfoMessageType")
  - RAR for responses to additional information requests

    (types "RequestAdditionalInfoResponseMessageType", "RequestAdditionalInfoResponsePushMessageType")
  - IAL for "isAlive" messages
  - FEM for "FunctionalErrorMessageType"
  - CAN for "CancellationMessageType"
- **[0-9]{15}** is a number consisting of 15 digits (an implementer can freely choose to either iterate this number each time a new message is created or choose a random number from a provided pool, as long as it ensures that this number has not been used before)

Please note that the numeric part of this technical identifier is defined as being unique on a national level and the technical identifier is made unique across all ECRIS messages by prefixing the numeric part with the Member State code.

Please note also that, depending on the type of message, different combinations of ID types for the "MessageMetaDataMessageID" and "MessageMetaDataResponseTo" are defined. This is done in order to enforce the rules defined in kinematics for message correlation and allowed message exchanges. As an example, the "CancelledMessageIDType" can only accept as values ID's of notification, notification response, request and requests response messages.

As an example, if Germany intends to send a notification message to Austria using ECRIS, the technical identifier used would look like "DE-AT-NOT-153264871659485" or "DE-AT-NOT-000000000015234".

Please note also that **no dummy value** must be used in this field.

- "MessageMetaDataTimestamp" carries the date and time of the moment when the message is **sent** by the Member State. This property is always mandatory; in particular no dummy value must be used in this field.

  The following rules must be observed for filling in this field properly:

  - the date and time up to the seconds should be inserted just before sending the message

- the date/time used should be the local date/time and the indication of the time-zone must be included

The expected values are thus of the following form: 2010-11-25T21:32:35+02:00

- "MessageMetaDataResponseTo" is a property that contains the technical identifier of the message to which this message is responding to (i.e. value of "MessageMetaDataMessageID" of a previous message). It allows thus correlating a message to one previously sent.

  In the ECRIS message exchanges, the messages are to be correlated in a **sequential** manner, meaning that the latest message must reference the previously **received** message **within the same transaction**.

  As a concrete example, let's assume that Lithuania sends a request message with a technical ID "LT-PL-REQ-123432653223112" to Poland. In response to that message, Poland sends a new message "PL-LT-RDL-321234177654353" containing the legal deadline for responding to the request. This message contains the "MessageMetaDataResponseTo" property filled with the technical identifier provided in the request of Lithuania: "LT-PL-REQ-123432653223112". Let's assume now that Poland finishes successfully the process and returns a message to Lithuania containing the requested information. This response message has for instance the unique technical identifier "PL-LT-RRS-32123417765435**4**". It also contains the "MessageMetaDataResponseTo" property filled with the technical identifier of the latest **received** message within this transaction, which is thus the request message that was sent by Lithuania.

  (Please note that the numeric part of the technical message identifier is not necessarily incremented by one unit and is not necessary assigned sequentially for several messages in a transaction.)

As described above, different extensions of the "AbstractMessageMetaDataType" have been defined so as to enforce the rules foreseen for message correlation and kinematics:

- **"MessageMetaDataType":** A message meta-data type used for all messages which are entry points for business processes and which do not correlate to previously send messages**.**
- **"CorrelatingMessageMetaDataType":** An extension to the abstract meta-data type used for all messages which are not entry points for business processes and which **MUST** correlate to a previously received message.
- **"CancellationMessageMetaDataType":** A meta-data type definition, used for transmitting cancellation messages.

  Attention: in the case of a cancellation message, the "MessageMetaDataResponseTo" property is filled with the technical identifier of the message **previously sent and being cancelled** within this transaction.

As a rule, the messages that initiate a new transaction (i.e. "NotificationMessageType", "NotificationMessagePushType", "RequestMessageType" and "RequestPushMessageType") use "MessageMetaDataType" as message meta-data. All subsequent messages use the "CorrelatingMessageMetaDataType". The "CancellationMessageMetaDataType" is only used for transmitting cancellation messages and thus the property "MessageMetaDataResponseTo" allows indicating only ID's for the types of messages that can actually be cancelled: request, request response, notification and notification response messages.

For illustrating the proper usage of the technical identifiers of messages and of the "MessageMetaDataResponseTo" property, let's take a complex kinematic in which Belgium sends a request to Poland (please note that the names of properties and numeric parts of the technical identifiers have been shortened for a better readability):

Request (BE > PL)	MessageID = BE-PL-REQ-123…456

Request deadline (PL > BE)	MessageID = PL-BE-RDL-987…321
	ResponseTo = BE-PL-REQ-123…456

Request for additional ID info	MessageID = PL-BE-RAI-987…322

(PL > BE)	ResponseTo = BE-PL-REQ-123…456

Response with additional ID info (BE > PL)	MessageID = BE-PL-RAR-123…463
	ResponseTo = PL-BE-RAI-987…322

Functional error (PL > BE)	MessageID = PL-BE-FEM-987…327
	ResponseTo = BE-PL-RAR-123…463

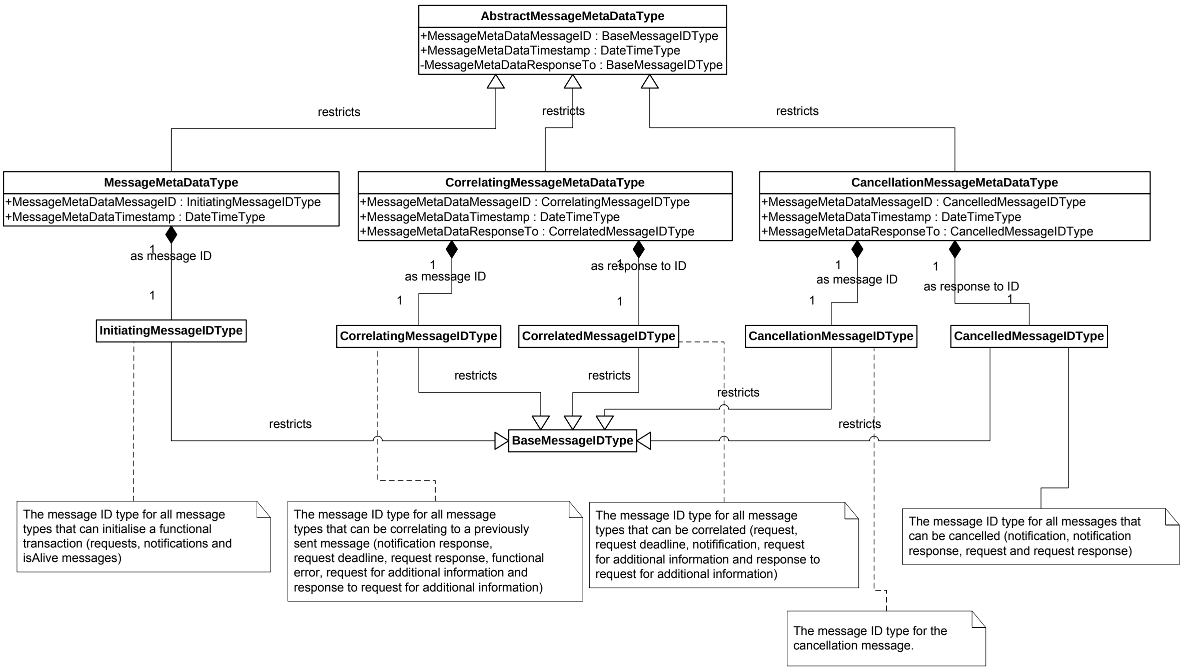
Response with additional ID info (BE > PL)	MessageID = BE-PL-RAR-123…466
	ResponseTo = PL-BE-FEM-987…327

Request response (PL > BE)	MessageID = PL-BE-RRS-987…331
	ResponseTo = BE-PL-RAR-123…466

Cancellation of request response (PL > BE)	MessageID = PL-BE-CAN-987…336
	ResponseTo = PL-BE-RRS-987…331

Request response (PL > BE)	MessageID = PL-BE-RRS-987…339
	ResponseTo = BE-PL-RAR-123…466

The following UML class diagram provides an overview of the structures to be used as message meta-data:


AbstractMessageMetaDataType
+MessageMetaDataMessageID : BaseMessageIDType
+MessageMetaDataTimestamp : DateTimeType
-MessageMetaDataResponseTo : BaseMessageIDType
restricts
restricts
restricts
MessageMetaDataType
+MessageMetaDataMessageID : InitiatingMessageIDType
+MessageMetaDataTimestamp : DateTimeType
CorrelatingMessageMetaDataType
+MessageMetaDataMessageID : CorrelatingMessageIDType
+MessageMetaDataTimestamp : DateTimeType
+MessageMetaDataResponseTo : CorrelatedMessageIDType
CancellationMessageMetaDataType
+MessageMetaDataMessageID : CancelledMessageIDType
+MessageMetaDataTimestamp : DateTimeType
+MessageMetaDataResponseTo : CancelledMessageIDType
1
as message ID
1
1
as message ID
1
1
as response to ID
1
1
as message ID
1
1
as response to ID
1
InitiatingMessageIDType
CorrelatingMessageIDType
CorrelatedMessageIDType
CancellationMessageIDType
CancelledMessageIDType
restricts
restricts
restricts
restricts
restricts
BaseMessageIDType

The message ID type for all message types that can initialise a functional transaction (requests, notifications and isAlive messages)

The message ID type for all message types that can be correlating to a previously sent message (notification response, request deadline, request response, functional error, request for additional information and response to request for additional information)

The message ID type for all message types that can be correlated (request, request deadline, notifification, request for additional information and response to request for additional information)

The message ID type for all messages that can be cancelled (notification, notification response, request and request response)

The message ID type for the cancellation message.

Figure 9 – Messages – Meta-Data types

Message Data

The message data part contains the functional information of the ECRIS message, and is thus clearly separated from the technical meta-data as explained previously.

The design principle that has been chosen for implementing this message data is to define a common base abstract type named "AbstractBusinessMessageDataType" and containing **all possible properties** of all business messages. All concrete message data types are then derived from this abstract type by **restriction**, only keeping the elements that are relevant to the specific message and redefining the cardinality as appropriate.

The main advantage is that the same structure and sequence of elements is enforced for all similar messages by their parent abstract type.

The following UML class diagram illustrates the "AbstractBusinessMessageDataType" element:

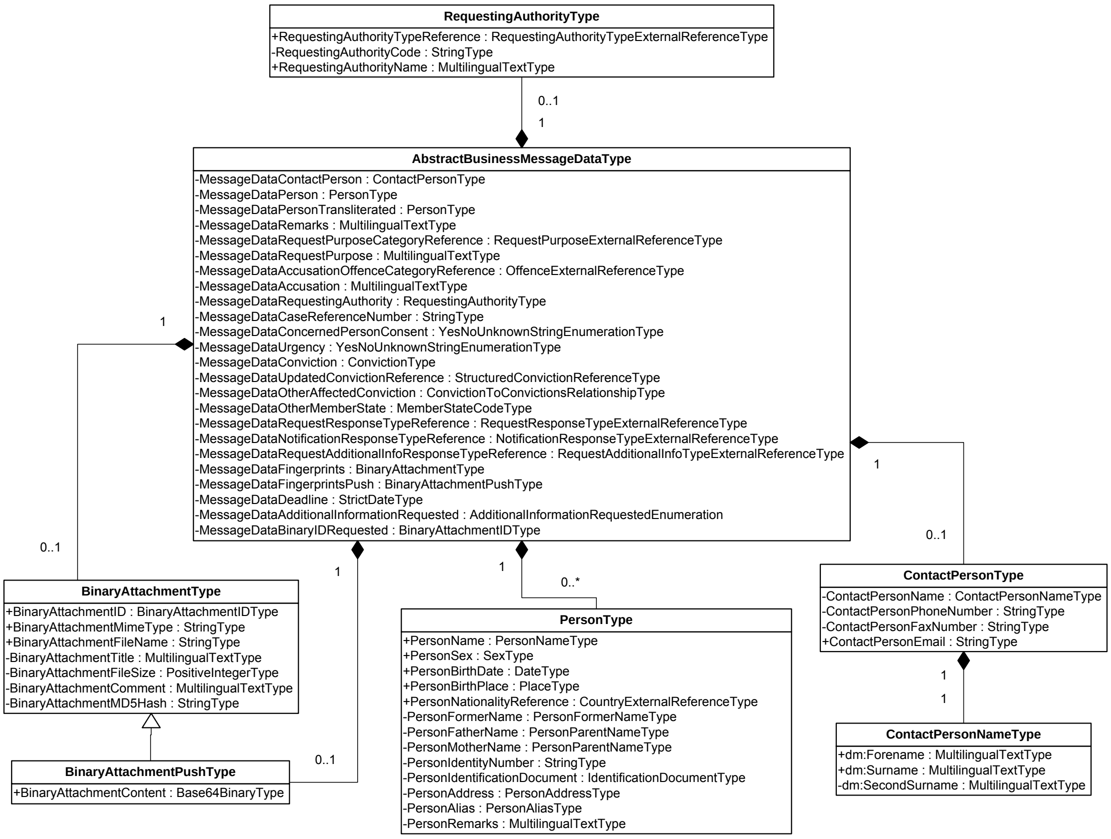

RequestingAuthorityType
+RequestingAuthorityTypeReference : RequestingAuthorityTypeExternalReferenceType
-RequestingAuthorityCode : StringType
+RequestingAuthorityName : MultilingualTextType
0..1
1
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
1
0..1
1
1
0..*
1
0..1
BinaryAttachmentType
+BinaryAttachmentID : BinaryAttachmentIDType
+BinaryAttachmentMimeType : StringType
+BinaryAttachmentFileName : StringType
-BinaryAttachmentTitle : MultilingualTextType
-BinaryAttachmentFileSize : PositiveIntegerType
-BinaryAttachmentComment : MultilingualTextType
-BinaryAttachmentMD5Hash : StringType
BinaryAttachmentPushType
+BinaryAttachmentContent : Base64BinaryType
0..1
PersonType
+PersonName : PersonNameType
+PersonSex : SexType
+PersonBirthDate : DateType
+PersonBirthPlace : PlaceType
+PersonNationalityReference : CountryExternalReferenceType
-PersonFormerName : PersonFormerNameType
-PersonFatherName : PersonParentNameType
-PersonMotherName : PersonParentNameType
-PersonIdentityNumber : StringType
-PersonIdentificationDocument : IdentificationDocumentType
-PersonAddress : PersonAddressType
-PersonAlias : PersonAliasType
-PersonRemarks : MultilingualTextType
ContactPersonType
-ContactPersonName : ContactPersonNameType
-ContactPersonPhoneNumber : StringType
-ContactPersonFaxNumber : StringType
+ContactPersonEmail : StringType
1
1
ContactPersonNameType
+dm:Forename : MultilingualTextType
+dm:Surname : MultilingualTextType
-dm:SecondSurname : MultilingualTextType

Figure 10 – Messages – "AbstractBusinessMessageDataType"

The following UML class diagrams illustrate the specific concrete message data types that are derived by restriction from the "AbstractBusinessMessageDataType":

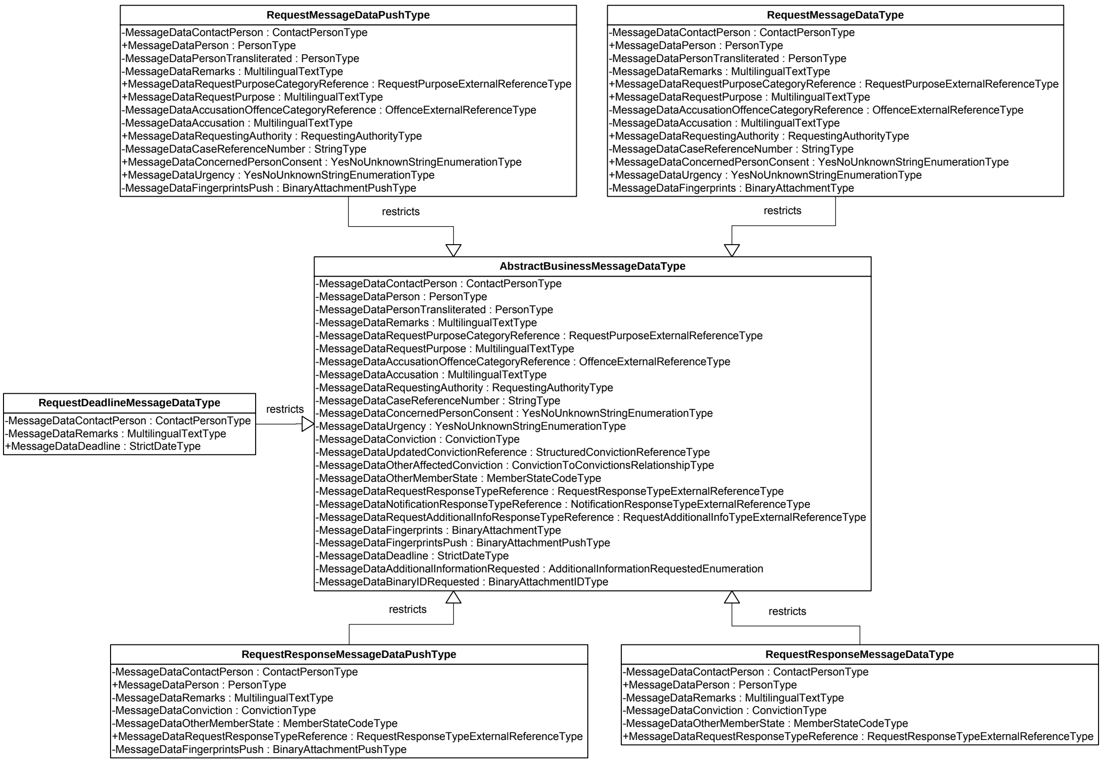

RequestMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
+MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
+MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
+MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
+MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
RequestMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
+MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
+MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
+MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
+MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataFingerprints : BinaryAttachmentType
restricts
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
RequestDeadlineMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataDeadline : StrictDateType
restricts
restricts
restricts
RequestResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataConviction : ConvictionType
-MessageDataOtherMemberState : MemberStateCodeType
+MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
RequestResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataConviction : ConvictionType
-MessageDataOtherMemberState : MemberStateCodeType
+MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType

Figure 11 – Message data types for "request" business process messages

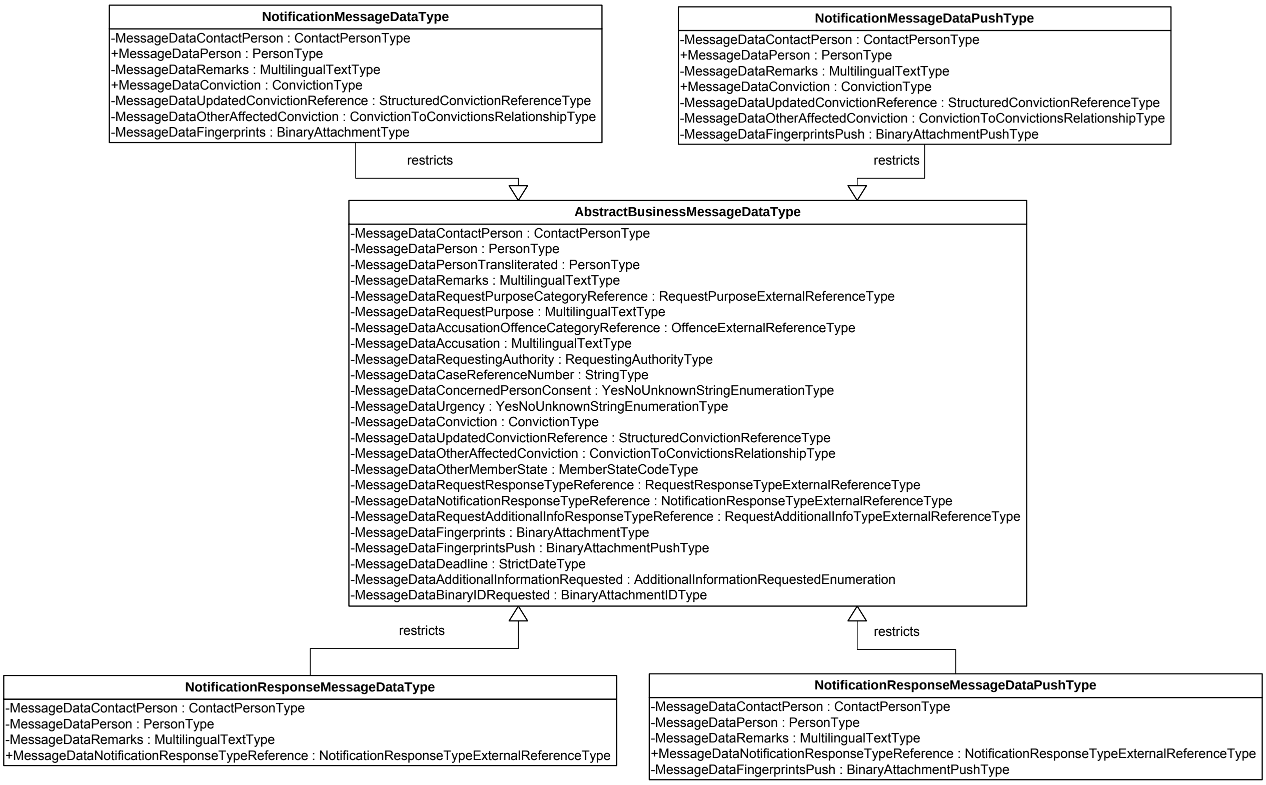

NotificationMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataFingerprints : BinaryAttachmentType
NotificationMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
restricts
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
restricts
restricts
NotificationResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
NotificationResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType

Figure 12 – Message data types for "notification" business process messages

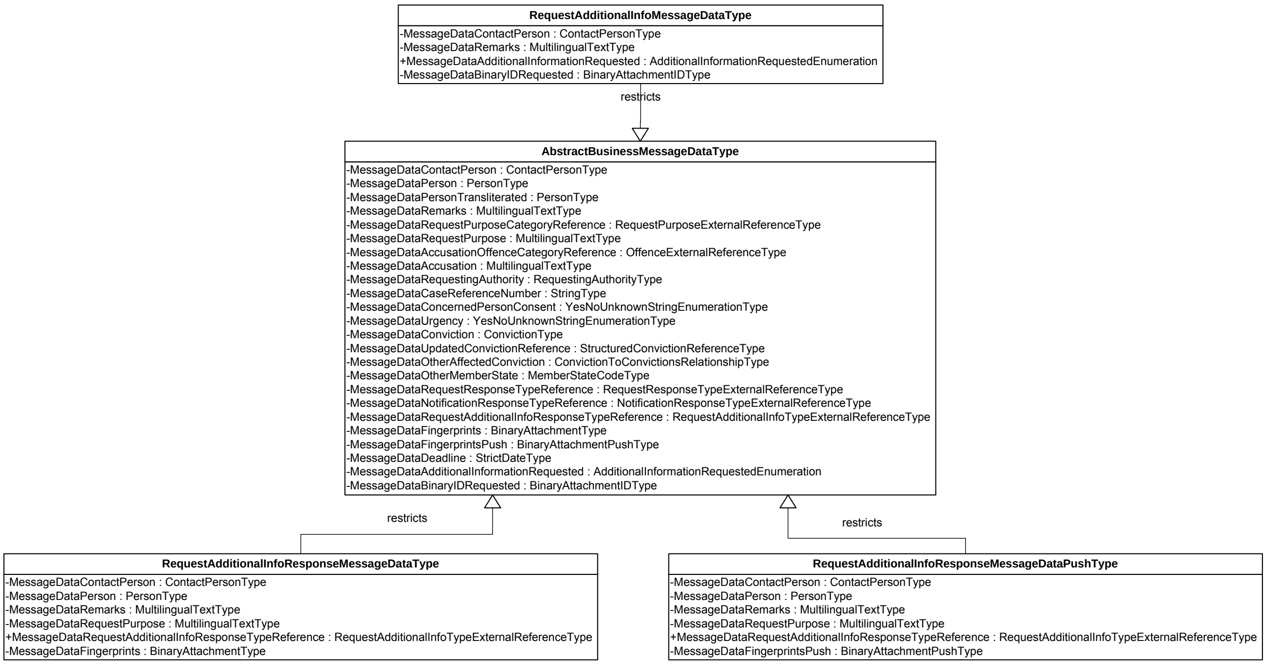

RequestAdditionalInfoMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataPersonTransliterated : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
restricts
restricts
RequestAdditionalInfoResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurpose : MultilingualTextType
+MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
RequestAdditionalInfoResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurpose : MultilingualTextType
+MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType

Figure 13 – Message data types for "search person" business sub-process messages

4.3 *Web Service* Operations

4.3.1 Style and Setting

The *web service* defined in the *WSDL* file follows the "Document" style using "Literal" as the "use" setting for the "soap:body" element. This style and setting is preferred to "RPC" style and "Encoding" setting mainly due to the fact that only this approach fulfils the requirements defined in the "Technical Architecture" document for performing synchronous structural validation of the message. Additionally using this style and setting ensures that the footprint of messages exchanged remains small because no encoding information is added in the message. Please note also that the *WS-I Basic Profile 1.1* specification, although not directly supported by the ECRIS detailed technical specifications, encourages the use of "Document" style using "Literal" setting and hints to the fact that the combination of "RPC" style with "Encoding" setting will most probably become deprecated in the future.

4.3.2 Transmission Primitive

Furthermore, the transmission primitive chosen for the operations (with the exception of the "IsAlive" operation) is the "one-way" primitive as defined in the *WSDL* 1.1 specification in chapter "2.4.1 – One-way Operation" (http://www.w3.org/TR/wsdl#_one-way). This transmission primitive is the most appropriate one given the fact that all operations (with the exception of "IsAlive") do not return a synchronous response message.

The main drawback of this primitive though is that, by definition, it does not foresee faults. However, since the definition of the *WSDL* specification, many implementers have realised that it is frequently necessary to be able to return faults in cases of errors, also when using operations that do not return response messages. Therefore, although not explicit in the *WSDL* 1.1 specification, most *web service* frameworks support returning faults when performing operations with input only parameters. Also, please note that in *WSDL* 2.0, which is the succeeding specification of WSDL 1.1, an appropriate transmission primitive named "robust-in-only" has been foreseen in order to accommodate such situations (see also http://www.w3.org/TR/wsdl20-adjuncts/#robust-in-only).

The solution adopted in the technical specifications is to define *web service* operations **without** faults on the *service contract* level. However, if a technical error occurs, implementers **must** return the standard "*SOAP* Fault" element and use either the standard fault codes or the custom fault codes that are defined in the common reference table "Faults". This way of working has been thoroughly tested with various technical platforms and is generally very well supported.

Please note that another alternative was initially considered and tested. Indeed the initial approach was to define custom fault types as part of the domain model and declare them accordingly in the *service contract* as possible fault return messages in the "one-way" operations. This approach would however not work in practice with some of the development platforms, namely "Axis1" and "Metro". Axis2 also encountered a problem processing the *service contract*, with the difference that in this case this shortcoming is considered a bug and has been fixed in the forthcoming version 1.6 (for more info on the problem, please check the following JIRA issue https://issues.apache.org/jira/browse/AXIS2-4408).

4.3.3 List of Operations

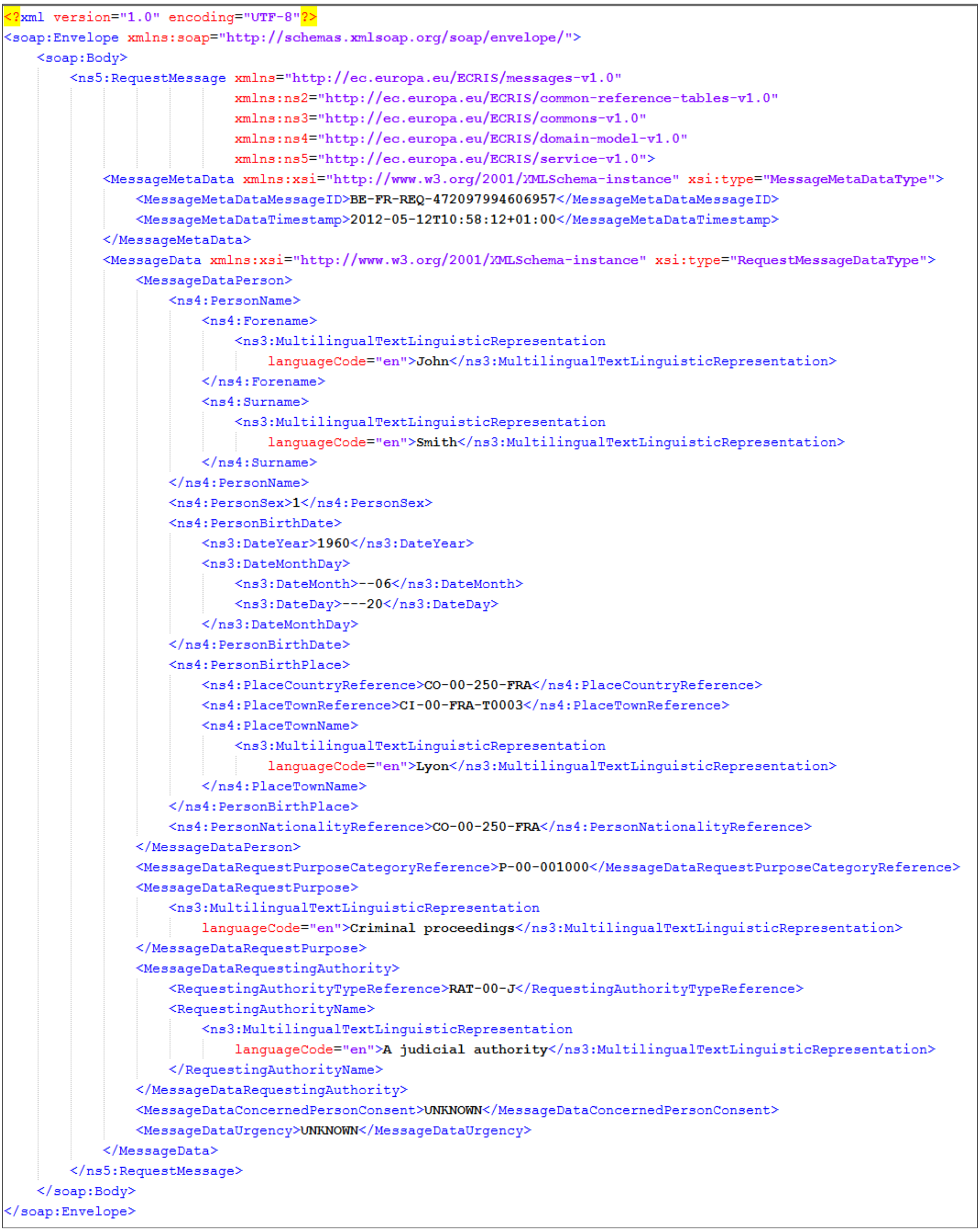
The following sections list the operations defined in the *WSDL* file.

For each operation a description as well as a sample message is presented. Please note that in the sample messages provided, optional elements are frequently left out for improving the readability of this document. In particular the samples used for showing exchanges including binary files are only included partially for the same reasons.

deliverRequest

This operation allows a Member State to transmit a request to another Member State for information and related data to be extracted from the criminal record of a person. It is the entry point that starts an instance of the "Request Criminal Record Information" business process defined in the "Business Analysis" document.

The following figure provides an example of the *SOAP* message:



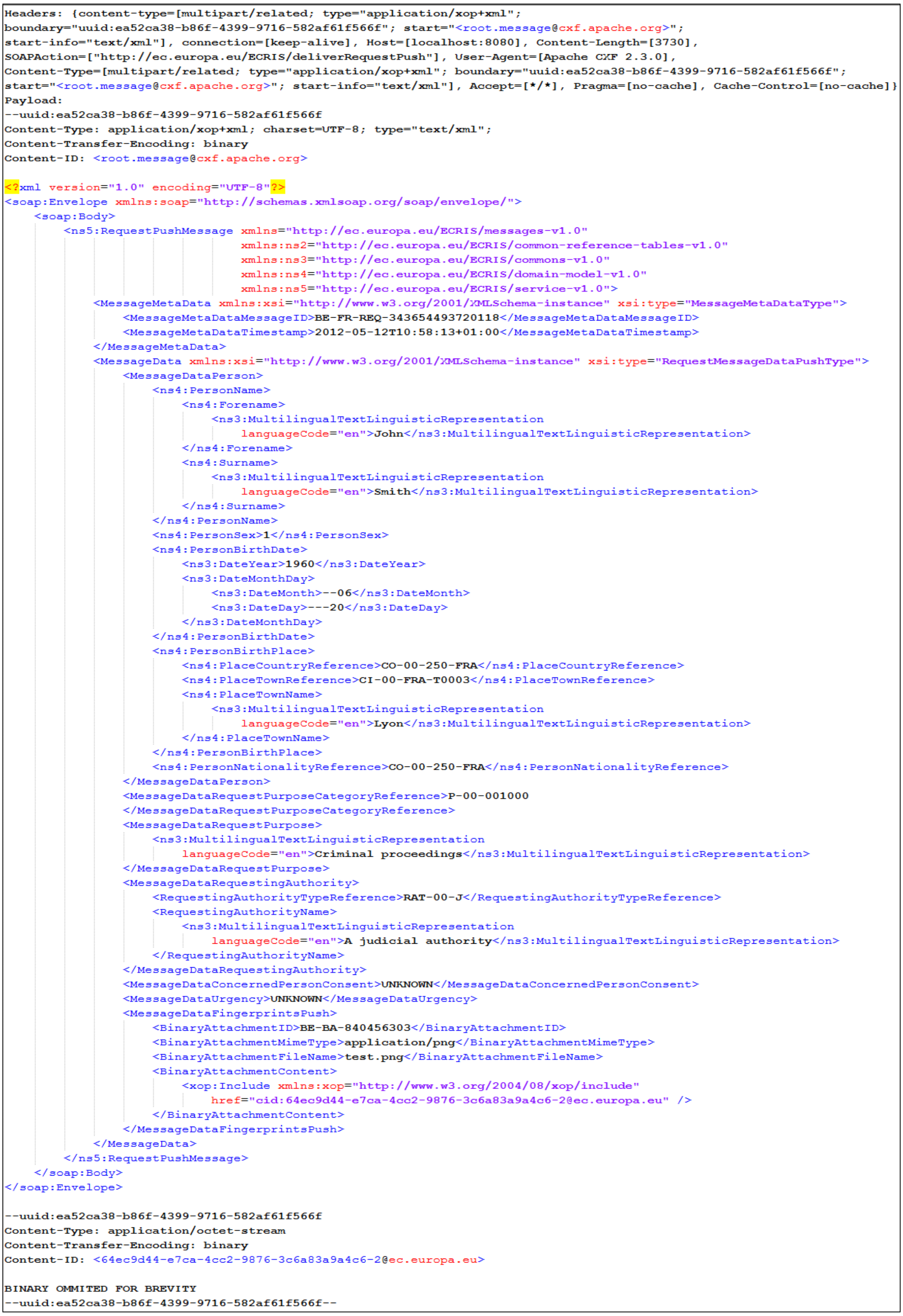
```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                            xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                            xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                            xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-REQ-472097994606957</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:12+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestPurposeCategoryReference>P-00-001000</MessageDataRequestPurposeCategoryReference>
                <MessageDataRequestPurpose>
                    <ns3:MultilingualTextLinguisticRepresentation
                        languageCode="en">Criminal proceedings</ns3:MultilingualTextLinguisticRepresentation>
                </MessageDataRequestPurpose>
                <MessageDataRequestingAuthority>
                    <RequestingAuthorityTypeReference>RAT-00-J</RequestingAuthorityTypeReference>
                    <RequestingAuthorityName>
                        <ns3:MultilingualTextLinguisticRepresentation
                            languageCode="en">A judicial authority</ns3:MultilingualTextLinguisticRepresentation>
                    </RequestingAuthorityName>
                </MessageDataRequestingAuthority>
                <MessageDataConcernedPersonConsent>UNKNOWN</MessageDataConcernedPersonConsent>
                <MessageDataUrgency>UNKNOWN</MessageDataUrgency>
            </MessageData>
        </ns5:RequestMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 14 – "RequestMessage" sample message

deliverRequestPush

This operation is identical to the "deliverRequest" operation but additionally supports sending the binary file as attachment to the "request" message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



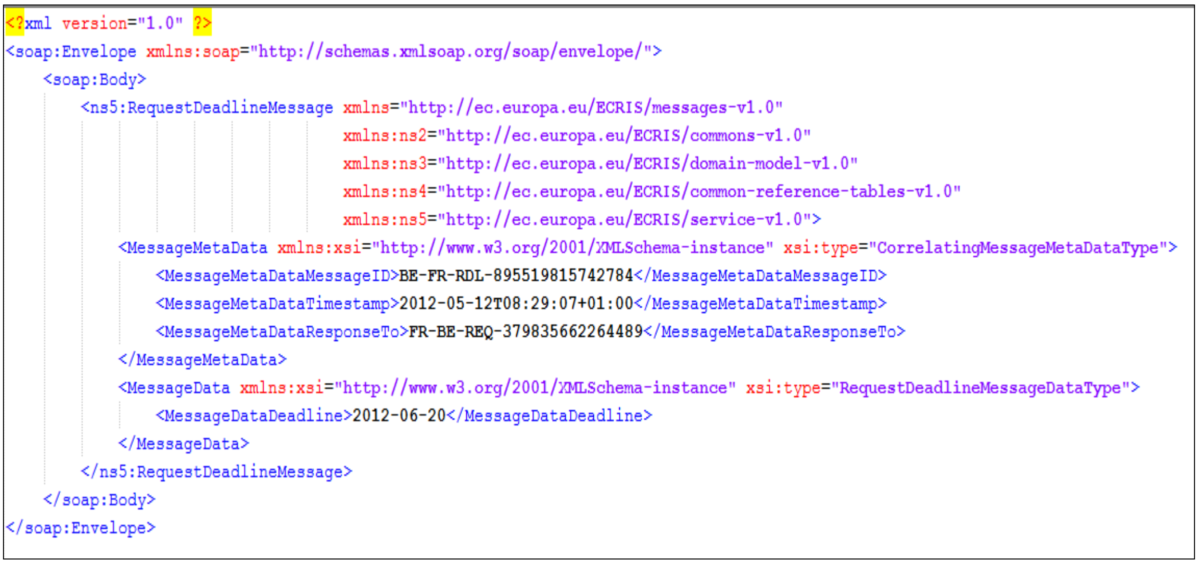
```
Headers: {content-type=[multipart/related; type="application/xop+xml";
boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], connection=[keep-alive], Host=[localhost:8080], Content-Length=[3730],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverRequestPush"], User-Agent=[Apache CXF 2.3.0],
Content-Type=[multipart/related; type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestPushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-REQ-343654493720118</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:13+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestMessageDataPushType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestPurposeCategoryReference>P-00-001000
                </MessageDataRequestPurposeCategoryReference>
                <MessageDataRequestPurpose>
                    <ns3:MultilingualTextLinguisticRepresentation
                        languageCode="en">Criminal proceedings</ns3:MultilingualTextLinguisticRepresentation>
                </MessageDataRequestPurpose>
                <MessageDataRequestingAuthority>
                    <RequestingAuthorityTypeReference>RAT-00-J</RequestingAuthorityTypeReference>
                    <RequestingAuthorityName>
                        <ns3:MultilingualTextLinguisticRepresentation
                            languageCode="en">A judicial authority</ns3:MultilingualTextLinguisticRepresentation>
                    </RequestingAuthorityName>
                </MessageDataRequestingAuthority>
                <MessageDataConcernedPersonConsent>UNKNOWN</MessageDataConcernedPersonConsent>
                <MessageDataUrgency>UNKNOWN</MessageDataUrgency>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-840456303</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:RequestPushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu>

BINARY OMMITED FOR BREVITY
--uuid:ea52ca38-b86f-4399-9716-582af61f566f--
```

Figure 15 – "RequestPushMessage" sample message

deliverRequestDeadline

This operation allows a Member State to transmit the legal deadline for the response to the requesting Member State. This operation can be used either directly after a request was received but also after receiving additional identification information through the "deliverRequestAdditionalInfo" operation.

The following figure provides an example of the *SOAP* message:



```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestDeadlineMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RDL-895519815742784</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:07+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-379835662264489</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestDeadlineMessageDataType">
                <MessageDataDeadline>2012-06-20</MessageDataDeadline>
            </MessageData>
        </ns5:RequestDeadlineMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 16 – "RequestDeadlineMessage" sample message

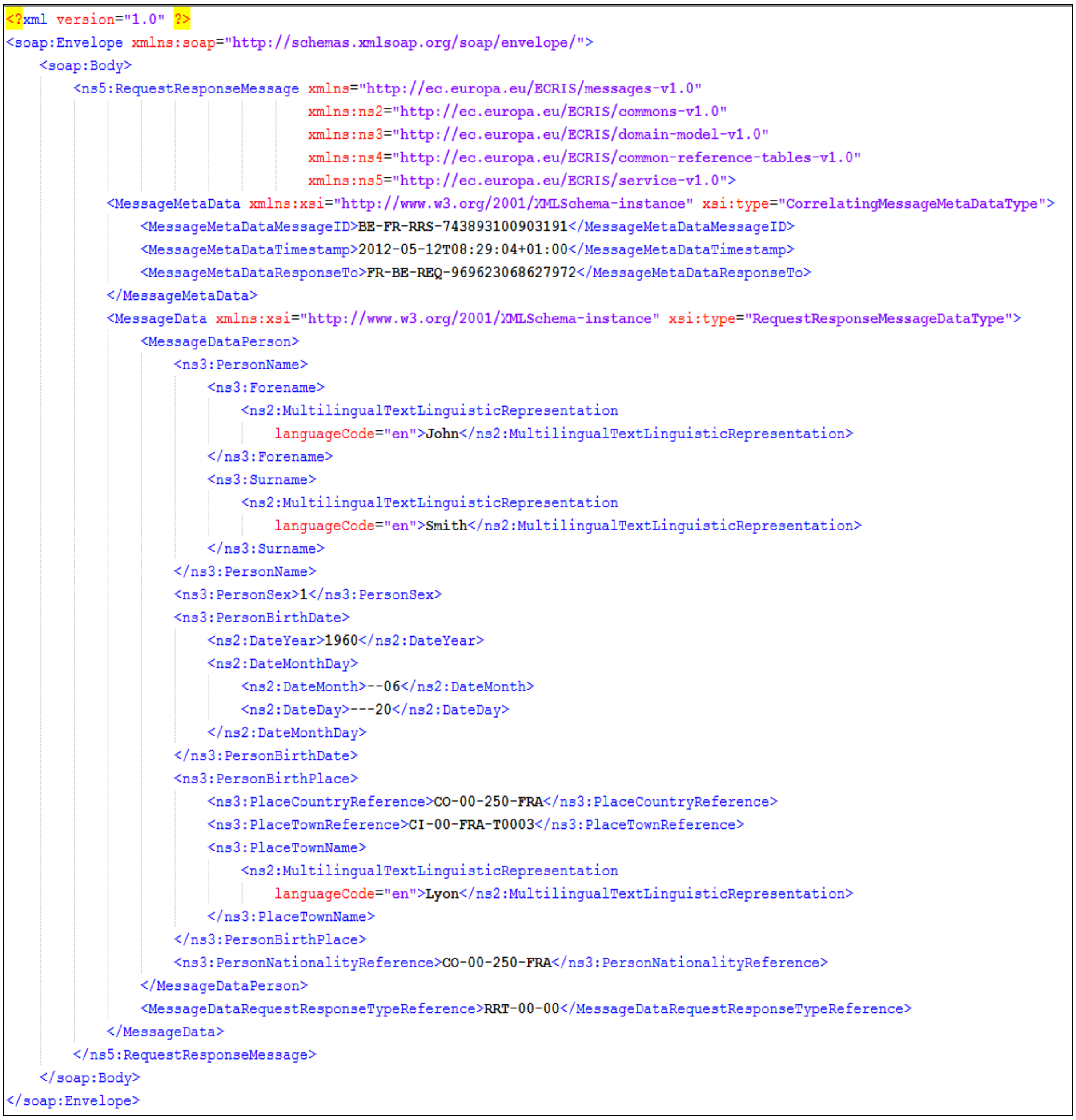
Please note that the legal deadline for the response to the request received is to be calculated implicitly in the local time of the **requested** Member State, and is to be considered **until the end of day, time 23:59:59**, of the transmitted deadline date. For example, in the depiction above, the deadline should be understood as 2012-06-20, until 23:59:59 hours GMT+1.

deliverRequestResponse

This operation allows a Member State to transmit the response to the request, which is the final operation in the "Request Criminal Record Information" business process.

It can contain: (1) information on convictions (if any) in the case that the request could be processed and that a single person matching the identification data has been found, (2) the request denial code if the request has been considered as not valid by the requested Member State or (3) a request problem code if a business issue occurred during the processing of the request.

The following figure provides an example containing no convictions of the *SOAP* message:



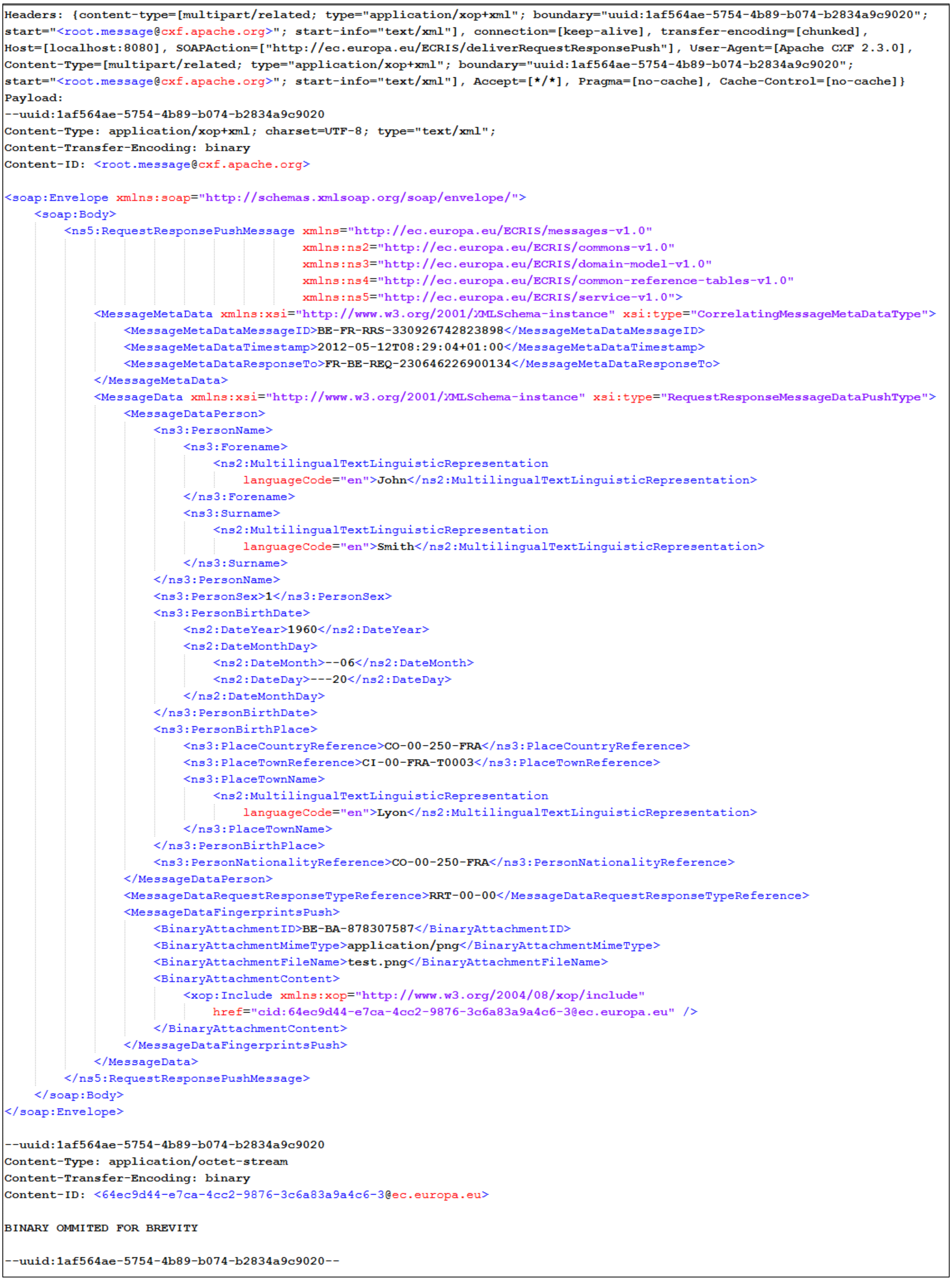
```xml
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RRS-743893100903191</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:04+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-969623068627972</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestResponseMessageDataType">
                <MessageDataPerson>
                    <ns3:PersonName>
                        <ns3:Forename>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:Forename>
                        <ns3:Surname>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:Surname>
                    </ns3:PersonName>
                    <ns3:PersonSex>1</ns3:PersonSex>
                    <ns3:PersonBirthDate>
                        <ns2:DateYear>1960</ns2:DateYear>
                        <ns2:DateMonthDay>
                            <ns2:DateMonth>--06</ns2:DateMonth>
                            <ns2:DateDay>---20</ns2:DateDay>
                        </ns2:DateMonthDay>
                    </ns3:PersonBirthDate>
                    <ns3:PersonBirthPlace>
                        <ns3:PlaceCountryReference>CO-00-250-FRA</ns3:PlaceCountryReference>
                        <ns3:PlaceTownReference>CI-00-FRA-T0003</ns3:PlaceTownReference>
                        <ns3:PlaceTownName>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:PlaceTownName>
                    </ns3:PersonBirthPlace>
                    <ns3:PersonNationalityReference>CO-00-250-FRA</ns3:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestResponseTypeReference>RRT-00-00</MessageDataRequestResponseTypeReference>
            </MessageData>
        </ns5:RequestResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 17 – "RequestResponseMessage" sample message

deliverRequestResponsePush

This operation is identical to the “deliverRequestResponse” operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the “Push” approach that has been in the “Technical Architecture” document).

The following figure provides an example of the *SOAP* message:



```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:1af564ae-5754-4b89-b074-b2834a9c9020";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], connection=[keep-alive], transfer-encoding=[chunked],
Host=[localhost:8080], SOAPAction=["http://ec.europa.eu/ECRIS/deliverRequestResponsePush"], User-Agent=[Apache CXF 2.3.0],
Content-Type=[multipart/related; type="application/xop+xml"; boundary="uuid:1af564ae-5754-4b89-b074-b2834a9c9020";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:1af564ae-5754-4b89-b074-b2834a9c9020
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestResponsePushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                        xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                        xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                        xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                        xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RRS-330926742823898</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:04+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-230646226900134</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestResponseMessageDataPushType">
                <MessageDataPerson>
                    <ns3:PersonName>
                        <ns3:Forename>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:Forename>
                        <ns3:Surname>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:Surname>
                    </ns3:PersonName>
                    <ns3:PersonSex>1</ns3:PersonSex>
                    <ns3:PersonBirthDate>
                        <ns2:DateYear>1960</ns2:DateYear>
                        <ns2:DateMonthDay>
                            <ns2:DateMonth>--06</ns2:DateMonth>
                            <ns2:DateDay>---20</ns2:DateDay>
                        </ns2:DateMonthDay>
                    </ns3:PersonBirthDate>
                    <ns3:PersonBirthPlace>
                        <ns3:PlaceCountryReference>CO-00-250-FRA</ns3:PlaceCountryReference>
                        <ns3:PlaceTownReference>CI-00-FRA-T0003</ns3:PlaceTownReference>
                        <ns3:PlaceTownName>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:PlaceTownName>
                    </ns3:PersonBirthPlace>
                    <ns3:PersonNationalityReference>CO-00-250-FRA</ns3:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestResponseTypeReference>RRT-00-00</MessageDataRequestResponseTypeReference>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-878307587</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-3@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:RequestResponsePushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:1af564ae-5754-4b89-b074-b2834a9c9020
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-3@ec.europa.eu>

BINARY OMMITED FOR BREVITY

--uuid:1af564ae-5754-4b89-b074-b2834a9c9020--
```

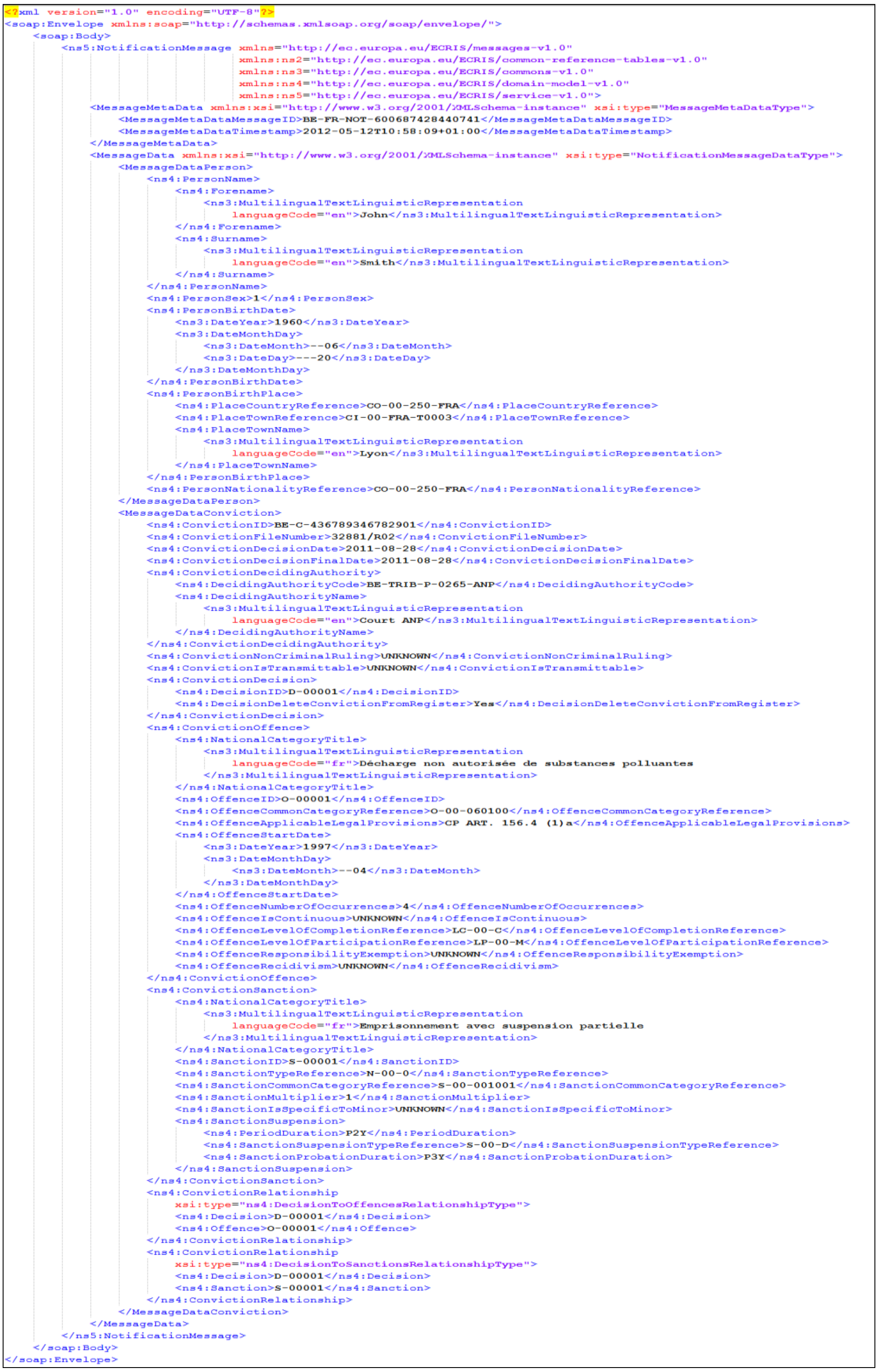
Figure 18 – "RequestResponsePushMessage" sample message

deliverNotification

This operation allows a Member State to transmit a notification to the Member State of the convicted person's nationality, carrying information on the conviction as well as information on subsequent alterations and deletions.

It is the entry point that starts an instance of the "Notify Convictions and Subsequent Changes" business process defined in the "Business Analysis" document.

The following figure provides an example of the *SOAP* message:



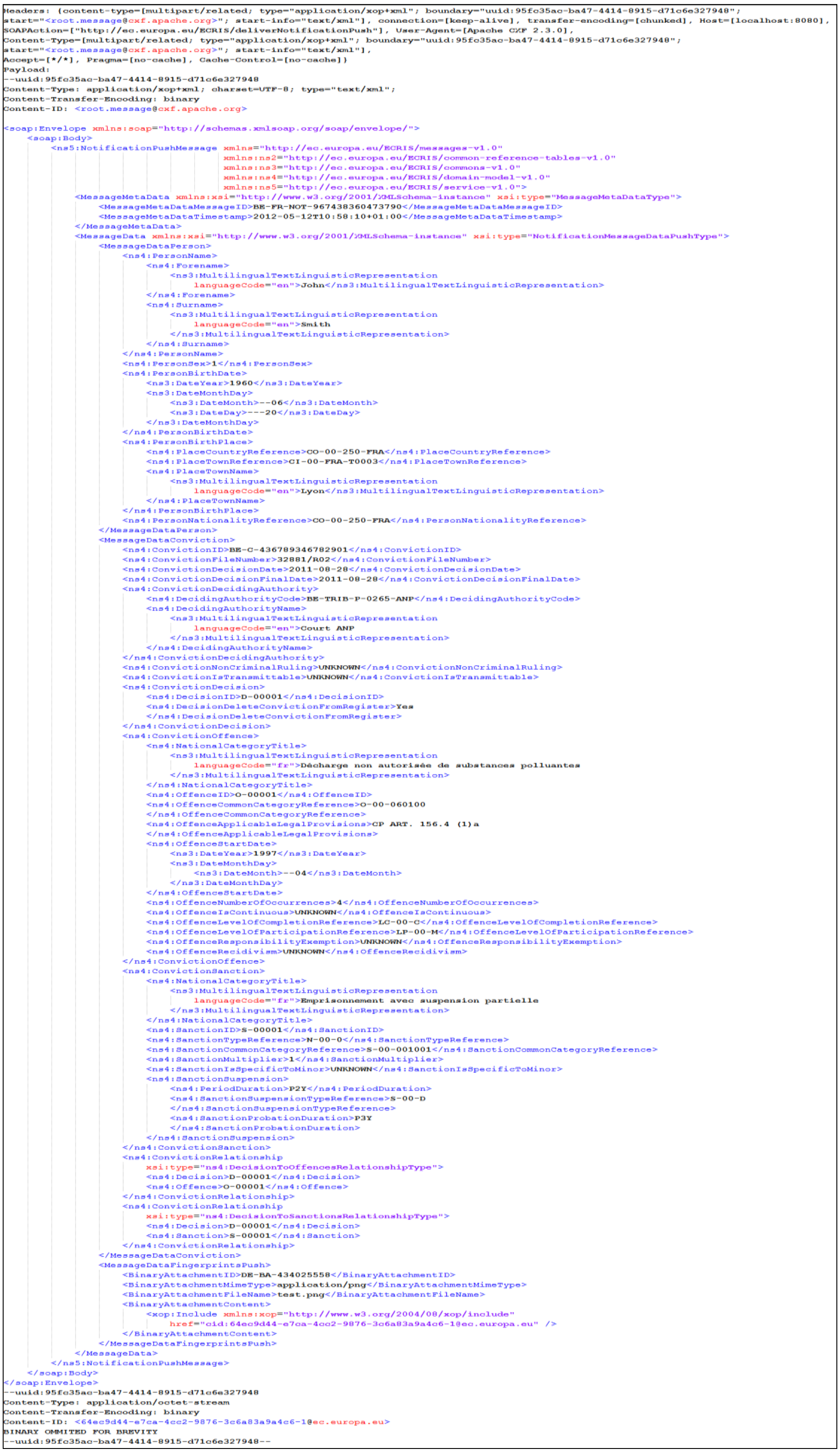
```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                 xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                 xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                 xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                 xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NOT-600687428440741</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:09+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataConviction>
                    <ns4:ConvictionID>BE-C-436789346782901</ns4:ConvictionID>
                    <ns4:ConvictionFileNumber>32881/R02</ns4:ConvictionFileNumber>
                    <ns4:ConvictionDecisionDate>2011-08-28</ns4:ConvictionDecisionDate>
                    <ns4:ConvictionDecisionFinalDate>2011-08-28</ns4:ConvictionDecisionFinalDate>
                    <ns4:ConvictionDecidingAuthority>
                        <ns4:DecidingAuthorityCode>BE-TRIB-P-0265-ANP</ns4:DecidingAuthorityCode>
                        <ns4:DecidingAuthorityName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Court ANP</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:DecidingAuthorityName>
                    </ns4:ConvictionDecidingAuthority>
                    <ns4:ConvictionNonCriminalRuling>UNKNOWN</ns4:ConvictionNonCriminalRuling>
                    <ns4:ConvictionIsTransmittable>UNKNOWN</ns4:ConvictionIsTransmittable>
                    <ns4:ConvictionDecision>
                        <ns4:DecisionID>D-00001</ns4:DecisionID>
                        <ns4:DecisionDeleteConvictionFromRegister>Yes</ns4:DecisionDeleteConvictionFromRegister>
                    </ns4:ConvictionDecision>
                    <ns4:ConvictionOffence>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Décharge non autorisée de substances polluantes
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:OffenceID>O-00001</ns4:OffenceID>
                        <ns4:OffenceCommonCategoryReference>O-00-060100</ns4:OffenceCommonCategoryReference>
                        <ns4:OffenceApplicableLegalProvisions>CP ART. 156.4 (1)a</ns4:OffenceApplicableLegalProvisions>
                        <ns4:OffenceStartDate>
                            <ns3:DateYear>1997</ns3:DateYear>
                            <ns3:DateMonthDay>
                                <ns3:DateMonth>--04</ns3:DateMonth>
                            </ns3:DateMonthDay>
                        </ns4:OffenceStartDate>
                        <ns4:OffenceNumberOfOccurrences>4</ns4:OffenceNumberOfOccurrences>
                        <ns4:OffenceIsContinuous>UNKNOWN</ns4:OffenceIsContinuous>
                        <ns4:OffenceLevelOfCompletionReference>LC-00-C</ns4:OffenceLevelOfCompletionReference>
                        <ns4:OffenceLevelOfParticipationReference>LP-00-M</ns4:OffenceLevelOfParticipationReference>
                        <ns4:OffenceResponsibilityExemption>UNKNOWN</ns4:OffenceResponsibilityExemption>
                        <ns4:OffenceRecidivism>UNKNOWN</ns4:OffenceRecidivism>
                    </ns4:ConvictionOffence>
                    <ns4:ConvictionSanction>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Emprisonnement avec suspension partielle
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:SanctionID>S-00001</ns4:SanctionID>
                        <ns4:SanctionTypeReference>N-00-0</ns4:SanctionTypeReference>
                        <ns4:SanctionCommonCategoryReference>S-00-001001</ns4:SanctionCommonCategoryReference>
                        <ns4:SanctionMultiplier>1</ns4:SanctionMultiplier>
                        <ns4:SanctionIsSpecificToMinor>UNKNOWN</ns4:SanctionIsSpecificToMinor>
                        <ns4:SanctionSuspension>
                            <ns4:PeriodDuration>P2Y</ns4:PeriodDuration>
                            <ns4:SanctionSuspensionTypeReference>S-00-D</ns4:SanctionSuspensionTypeReference>
                            <ns4:SanctionProbationDuration>P3Y</ns4:SanctionProbationDuration>
                        </ns4:SanctionSuspension>
                    </ns4:ConvictionSanction>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToOffencesRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Offence>O-00001</ns4:Offence>
                    </ns4:ConvictionRelationship>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToSanctionsRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Sanction>S-00001</ns4:Sanction>
                    </ns4:ConvictionRelationship>
                </MessageDataConviction>
            </MessageData>
        </ns5:NotificationMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 19 – “NotificationMessage” sample message

deliverNotificationPush

This operation is identical to the "deliverNotification" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



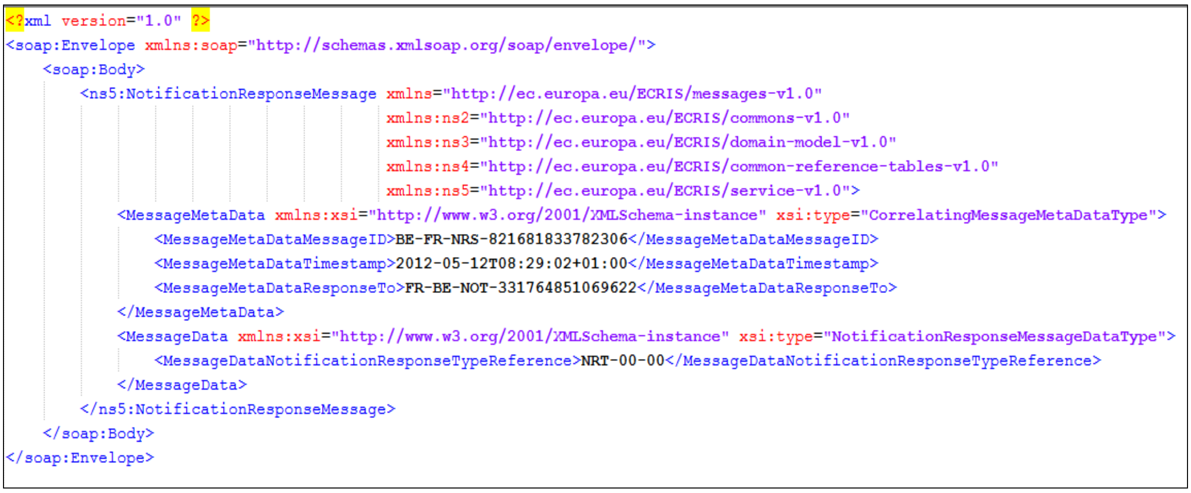
```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:95fc35ac-ba47-4414-8915-d71c6e327948";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], connection=[keep-alive], transfer-encoding=[chunked], Host=[localhost:8080],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverNotificationPush"], User-Agent=[Apache CXF 2.3.0],
Content-Type=[multipart/related; type="application/xop+xml"; boundary="uuid:95fc35ac-ba47-4414-8915-d71c6e327948";
start="<root.message@cxf.apache.org>"; start-info="text/xml"],
Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationPushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                     xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                     xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                     xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                     xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NOT-967438360473790</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:10+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationMessageDataPushType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataConviction>
                    <ns4:ConvictionID>BE-C-436789346782901</ns4:ConvictionID>
                    <ns4:ConvictionFileNumber>32881/R02</ns4:ConvictionFileNumber>
                    <ns4:ConvictionDecisionDate>2011-08-28</ns4:ConvictionDecisionDate>
                    <ns4:ConvictionDecisionFinalDate>2011-08-28</ns4:ConvictionDecisionFinalDate>
                    <ns4:ConvictionDecidingAuthority>
                        <ns4:DecidingAuthorityCode>BE-TRIB-P-0265-ANP</ns4:DecidingAuthorityCode>
                        <ns4:DecidingAuthorityName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Court ANP
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:DecidingAuthorityName>
                    </ns4:ConvictionDecidingAuthority>
                    <ns4:ConvictionNonCriminalRuling>UNKNOWN</ns4:ConvictionNonCriminalRuling>
                    <ns4:ConvictionIsTransmittable>UNKNOWN</ns4:ConvictionIsTransmittable>
                    <ns4:ConvictionDecision>
                        <ns4:DecisionID>D-00001</ns4:DecisionID>
                        <ns4:DecisionDeleteConvictionFromRegister>Yes
                        </ns4:DecisionDeleteConvictionFromRegister>
                    </ns4:ConvictionDecision>
                    <ns4:ConvictionOffence>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Décharge non autorisée de substances polluantes
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:OffenceID>O-00001</ns4:OffenceID>
                        <ns4:OffenceCommonCategoryReference>O-00-060100
                        </ns4:OffenceCommonCategoryReference>
                        <ns4:OffenceApplicableLegalProvisions>CP ART. 156.4 (1)a
                        </ns4:OffenceApplicableLegalProvisions>
                        <ns4:OffenceStartDate>
                            <ns3:DateYear>1997</ns3:DateYear>
                            <ns3:DateMonthDay>
                                <ns3:DateMonth>--04</ns3:DateMonth>
                            </ns3:DateMonthDay>
                        </ns4:OffenceStartDate>
                        <ns4:OffenceNumberOfOccurrences>4</ns4:OffenceNumberOfOccurrences>
                        <ns4:OffenceIsContinuous>UNKNOWN</ns4:OffenceIsContinuous>
                        <ns4:OffenceLevelOfCompletionReference>LC-00-C</ns4:OffenceLevelOfCompletionReference>
                        <ns4:OffenceLevelOfParticipationReference>LP-00-M</ns4:OffenceLevelOfParticipationReference>
                        <ns4:OffenceResponsibilityExemption>UNKNOWN</ns4:OffenceResponsibilityExemption>
                        <ns4:OffenceRecidivism>UNKNOWN</ns4:OffenceRecidivism>
                    </ns4:ConvictionOffence>
                    <ns4:ConvictionSanction>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Emprisonnement avec suspension partielle
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:SanctionID>S-00001</ns4:SanctionID>
                        <ns4:SanctionTypeReference>N-00-0</ns4:SanctionTypeReference>
                        <ns4:SanctionCommonCategoryReference>S-00-001001</ns4:SanctionCommonCategoryReference>
                        <ns4:SanctionMultiplier>1</ns4:SanctionMultiplier>
                        <ns4:SanctionIsSpecificToMinor>UNKNOWN</ns4:SanctionIsSpecificToMinor>
                        <ns4:SanctionSuspension>
                            <ns4:PeriodDuration>P2Y</ns4:PeriodDuration>
                            <ns4:SanctionSuspensionTypeReference>S-00-D
                            </ns4:SanctionSuspensionTypeReference>
                            <ns4:SanctionProbationDuration>P3Y
                            </ns4:SanctionProbationDuration>
                        </ns4:SanctionSuspension>
                    </ns4:ConvictionSanction>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToOffencesRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Offence>O-00001</ns4:Offence>
                    </ns4:ConvictionRelationship>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToSanctionsRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Sanction>S-00001</ns4:Sanction>
                    </ns4:ConvictionRelationship>
                </MessageDataConviction>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>DE-BA-434025558</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-1@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:NotificationPushMessage>
    </soap:Body>
</soap:Envelope>
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-1@ec.europa.eu>
BINARY OMMITED FOR BREVITY
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948--
```

Figure 20 – “NotificationPushMessage” sample message

deliverNotificationResponse

This operation allows a Member State to inform the convicting Member State of the outcome of the processing of the notification. It is the final operation in the "Notify Convictions and Subsequent Changes" business process. The message can contain: (1) a success message and optionally the identification information under which the notification has been processed or (2) a problem code if a business issue occurred during the processing of the notification.

The following figure provides an example of the *SOAP* message:



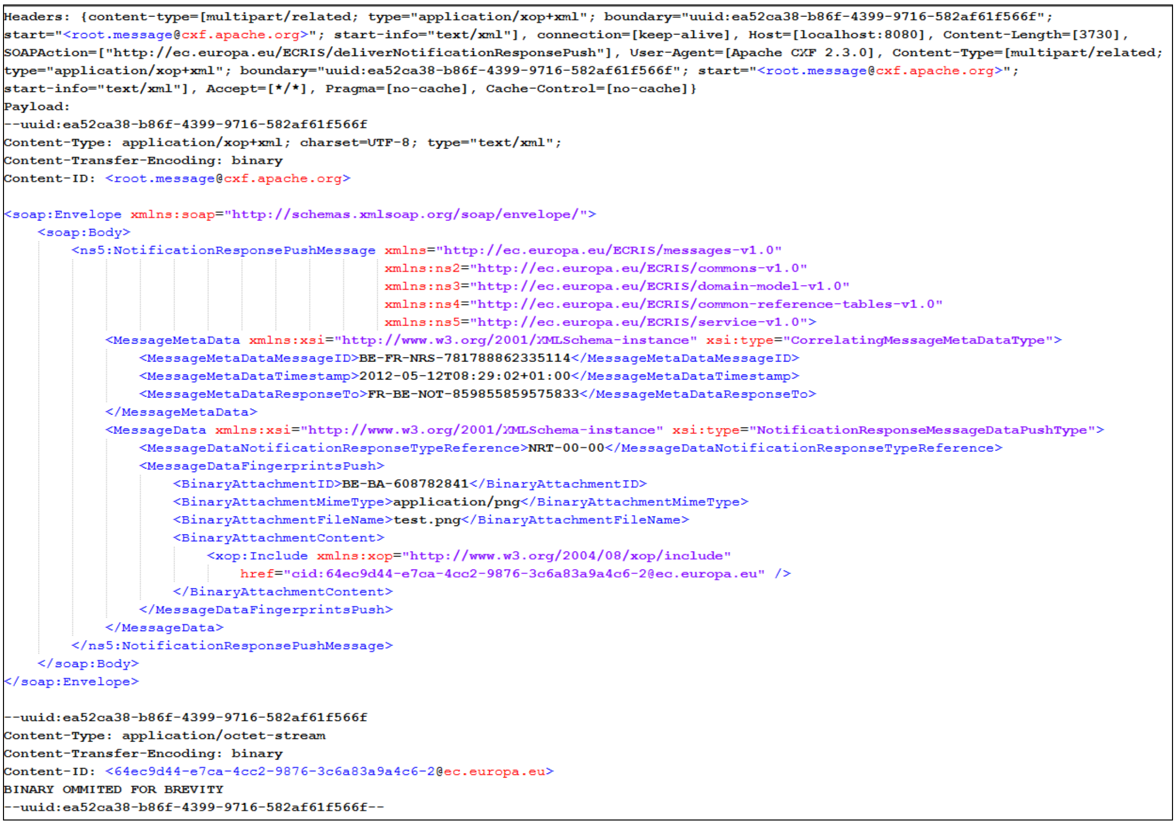
```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                         xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                         xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                         xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                         xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NRS-821681833782306</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:02+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-NOT-331764851069622</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationResponseMessageDataType">
                <MessageDataNotificationResponseTypeReference>NRT-00-00</MessageDataNotificationResponseTypeReference>
            </MessageData>
        </ns5:NotificationResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 21 – "NotificationResponseMessage" sample message

deliverNotificationResponsePush

This operation is identical to the "deliverNotificationResponse" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



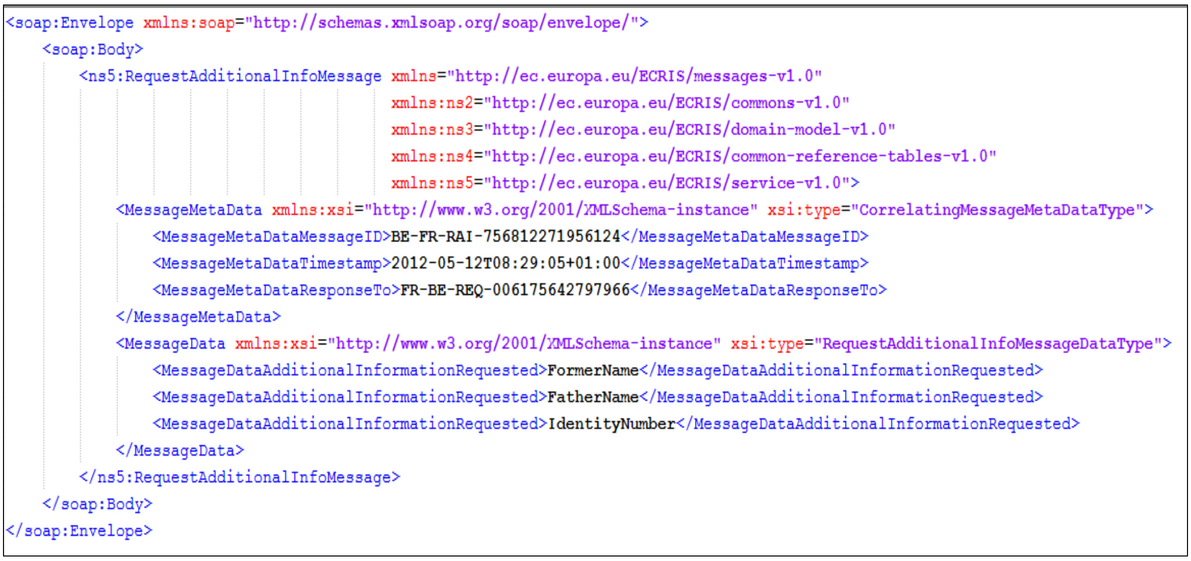
```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], connection=[keep-alive], Host=[localhost:8080], Content-Length=[3730],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverNotificationResponsePush"], User-Agent=[Apache CXF 2.3.0], Content-Type=[multipart/related;
type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationResponsePushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                             xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                             xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                             xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                             xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NRS-781788862335114</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:02+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-NOT-859855859575833</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationResponseMessageDataPushType">
                <MessageDataNotificationResponseTypeReference>NRT-00-00</MessageDataNotificationResponseTypeReference>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-608782841</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:NotificationResponsePushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu>
BINARY OMMITED FOR BREVITY
--uuid:ea52ca38-b86f-4399-9716-582af61f566f--
```

Figure 22 – "NotificationResponsePushMessage" sample message

deliverRequestAdditionalInfo

This operation is used by the Member State performing the search of a person for asking additional identification information to the requesting/convicting Member State.

The following figure provides an example of the *SOAP* message:



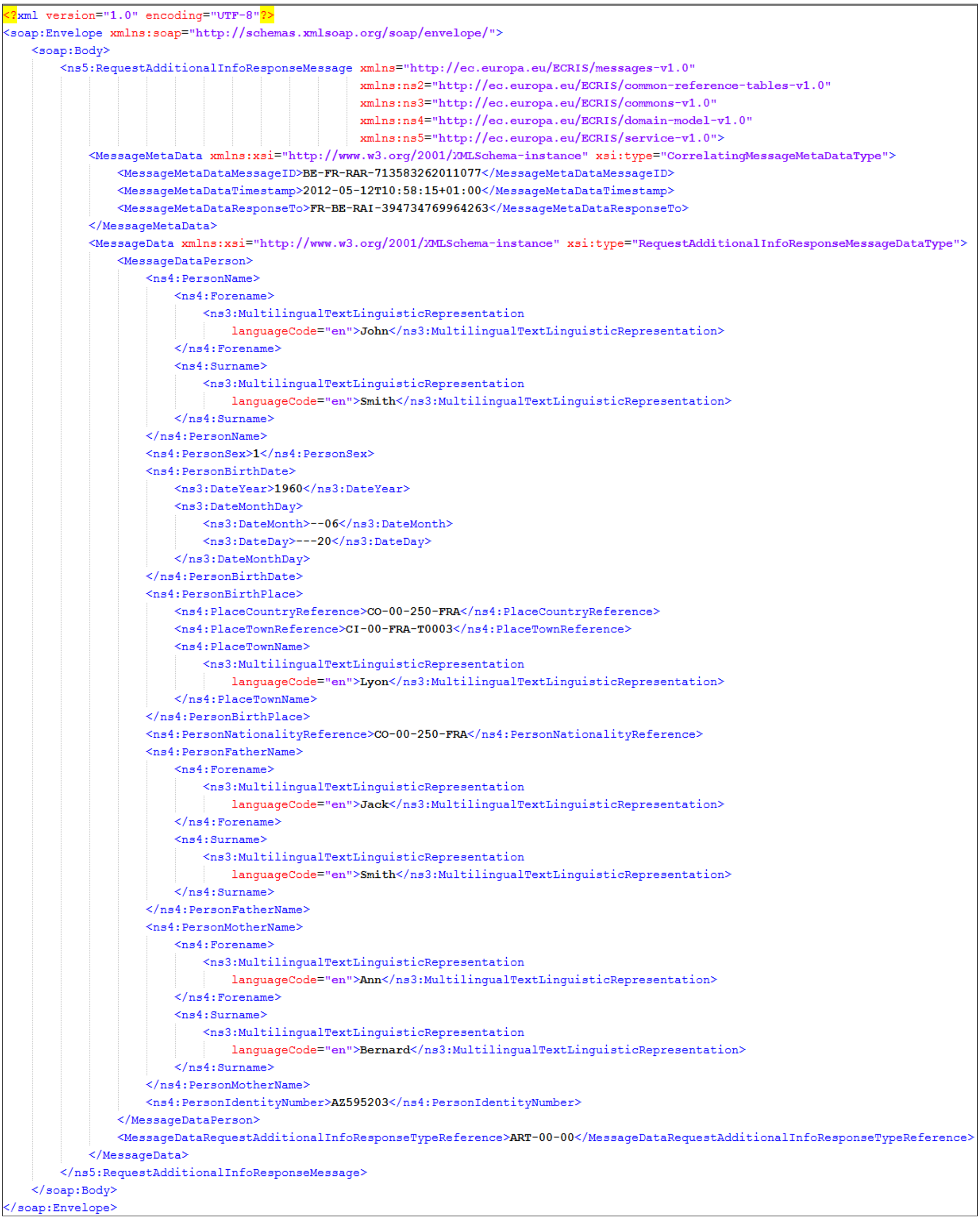
```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                          xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                          xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                          xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                          xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAI-756812271956124</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:05+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-006175642797966</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoMessageDataType">
                <MessageDataAdditionalInformationRequested>FormerName</MessageDataAdditionalInformationRequested>
                <MessageDataAdditionalInformationRequested>FatherName</MessageDataAdditionalInformationRequested>
                <MessageDataAdditionalInformationRequested>IdentityNumber</MessageDataAdditionalInformationRequested>
            </MessageData>
        </ns5:RequestAdditionalInfoMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 23 – "RequestAdditionalInfoMessage" sample message

deliverRequestAdditionalInfoResponse

This operation allows a Member State to respond to the "request additional info" message, either with updated personal identification data or with an indication that no additional information is available.

The following figure provides an example of the *SOAP* message:



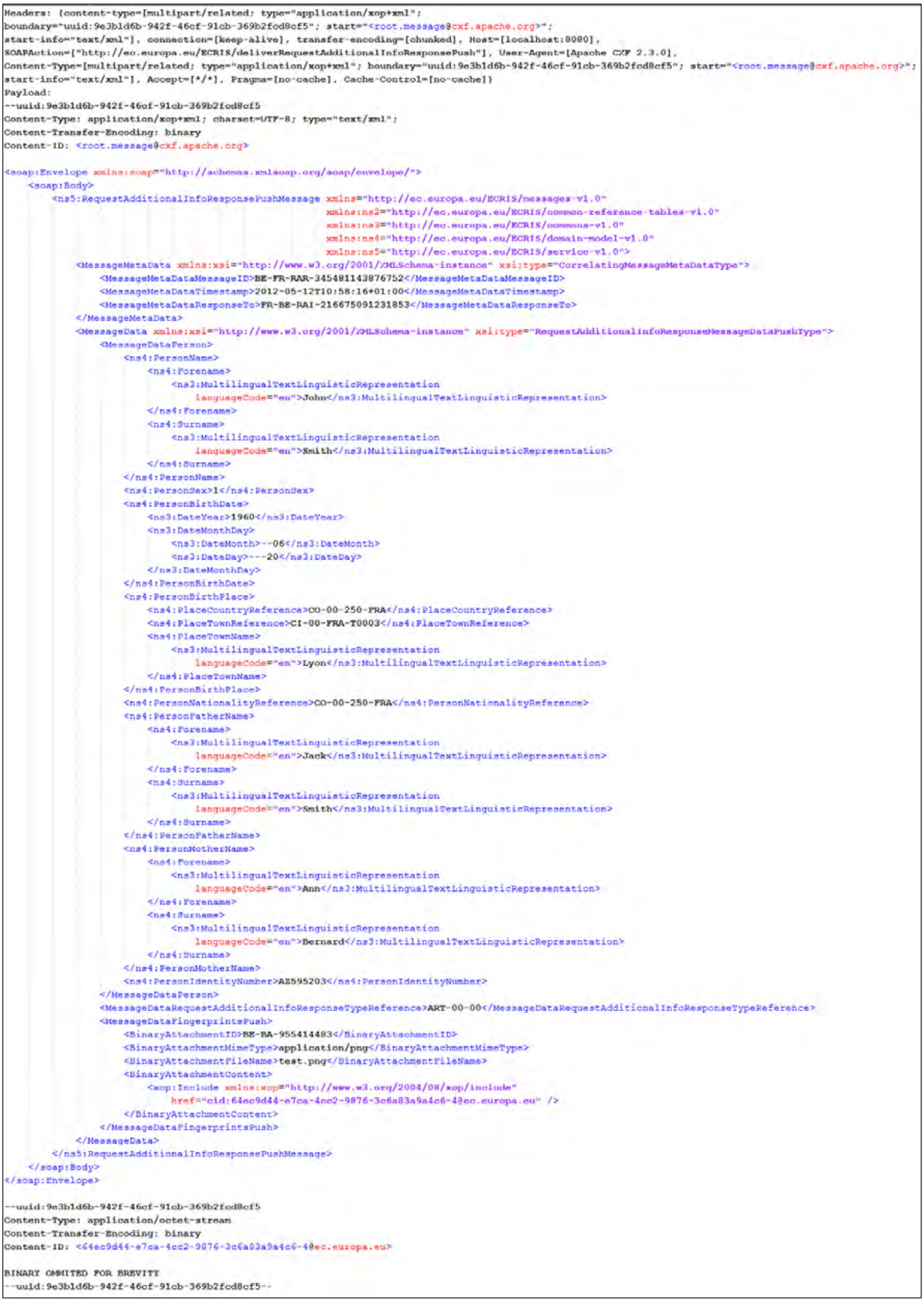
```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                                  xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                                  xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                                  xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                                  xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAR-713583262011077</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:15+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-RAI-394734769964263</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoResponseMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                    <ns4:PersonFatherName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Jack</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonFatherName>
                    <ns4:PersonMotherName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Ann</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Bernard</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonMotherName>
                    <ns4:PersonIdentityNumber>AZ595203</ns4:PersonIdentityNumber>
                </MessageDataPerson>
                <MessageDataRequestAdditionalInfoResponseTypeReference>ART-00-00</MessageDataRequestAdditionalInfoResponseTypeReference>
            </MessageData>
        </ns5:RequestAdditionalInfoResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 24 – “RequestAdditionalInfoResponseMessage” sample message

deliverRequestAdditionalInfoResponsePush

This operation is identical to the "deliverRequestAdditionalInfoResponse" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



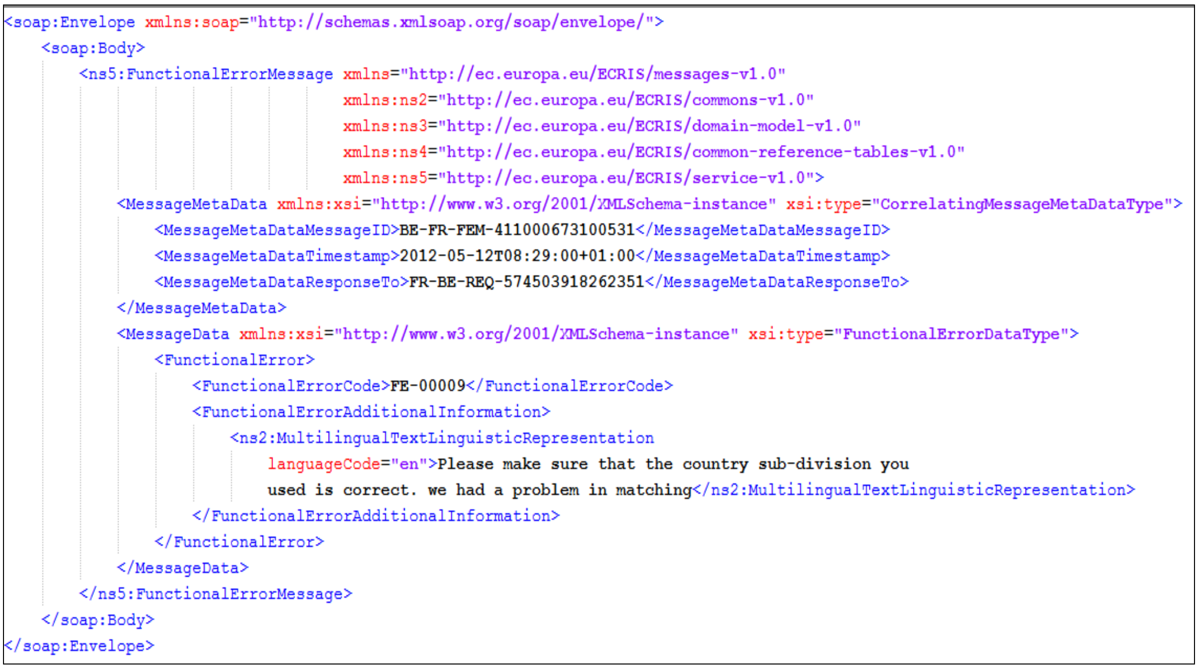
```
Headers: {content-type=[multipart/related; type="application/xop+xml";
boundary="uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], connection=[keep-alive], transfer-encoding=[chunked], Host=[localhost:8080],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverRequestAdditionalInfoResponsePush"], User-Agent=[Apache CXF 2.3.0],
Content-Type=[multipart/related; type="application/xop+xml"; boundary="uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoResponsePushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                                      xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                                      xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                                      xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                                      xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAR-345481143876752</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:16+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-RAI-216675091231853</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoResponseMessageDataPushType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">John</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                    <ns4:PersonFatherName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Jack</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Smith</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonFatherName>
                    <ns4:PersonMotherName>
                        <ns4:Forename>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Ann</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Forename>
                        <ns4:Surname>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Bernard</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:Surname>
                    </ns4:PersonMotherName>
                    <ns4:PersonIdentityNumber>AZ595203</ns4:PersonIdentityNumber>
                </MessageDataPerson>
                <MessageDataRequestAdditionalInfoResponseTypeReference>ART-00-00</MessageDataRequestAdditionalInfoResponseTypeReference>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-955414483</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-4@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:RequestAdditionalInfoResponsePushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-4@ec.europa.eu>

BINARY OMMITED FOR BREVITY
--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5--
```

Figure 25 – “RequestAdditionalInfoResponsePushMessage” sample message

deliverFunctionalError

This operation allows a Member State to provide information to a sending Member State on a functional error that occurred during the processing of a previous ECRIS message. It can be used at any time once an instance of a business process started and before it finishes.

The following figure provides an example of the *SOAP* message:



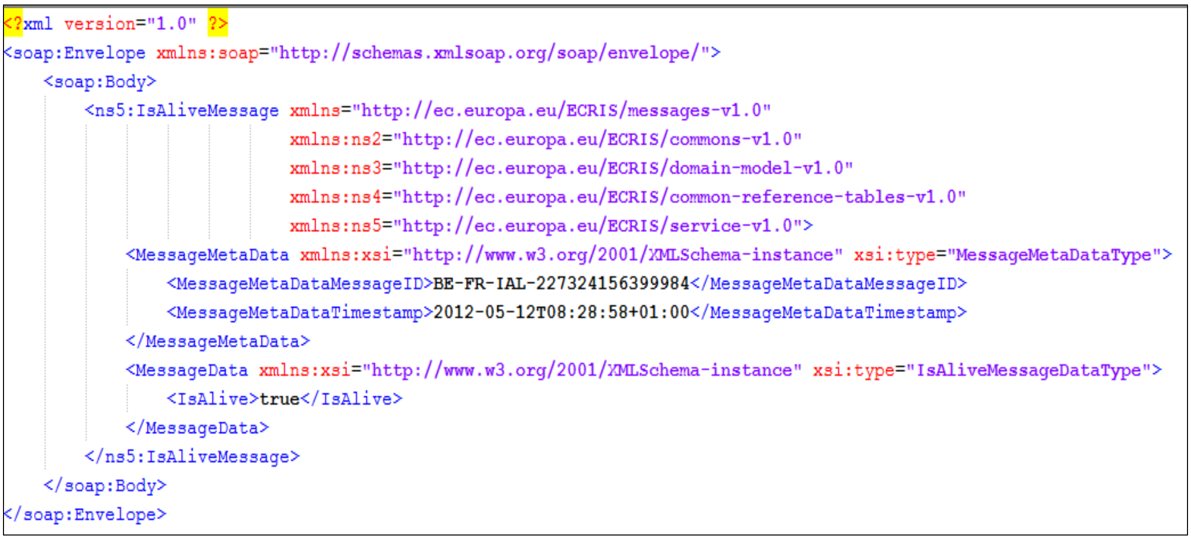
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:FunctionalErrorMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-FEM-411000673100531</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:00+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-574503918262351</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="FunctionalErrorDataType">
                <FunctionalError>
                    <FunctionalErrorCode>FE-00009</FunctionalErrorCode>
                    <FunctionalErrorAdditionalInformation>
                        <ns2:MultilingualTextLinguisticRepresentation
                            languageCode="en">Please make sure that the country sub-division you
                            used is correct. we had a problem in matching</ns2:MultilingualTextLinguisticRepresentation>
                    </FunctionalErrorAdditionalInformation>
                </FunctionalError>
            </MessageData>
        </ns5:FunctionalErrorMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 26 – "FunctionalErrorMessage" sample message

isAlive

The "isAlive" operation allows checking whether another ECRIS *web service* endpoint is properly responding to calls.

The following figure provides an example of the *SOAP* message:



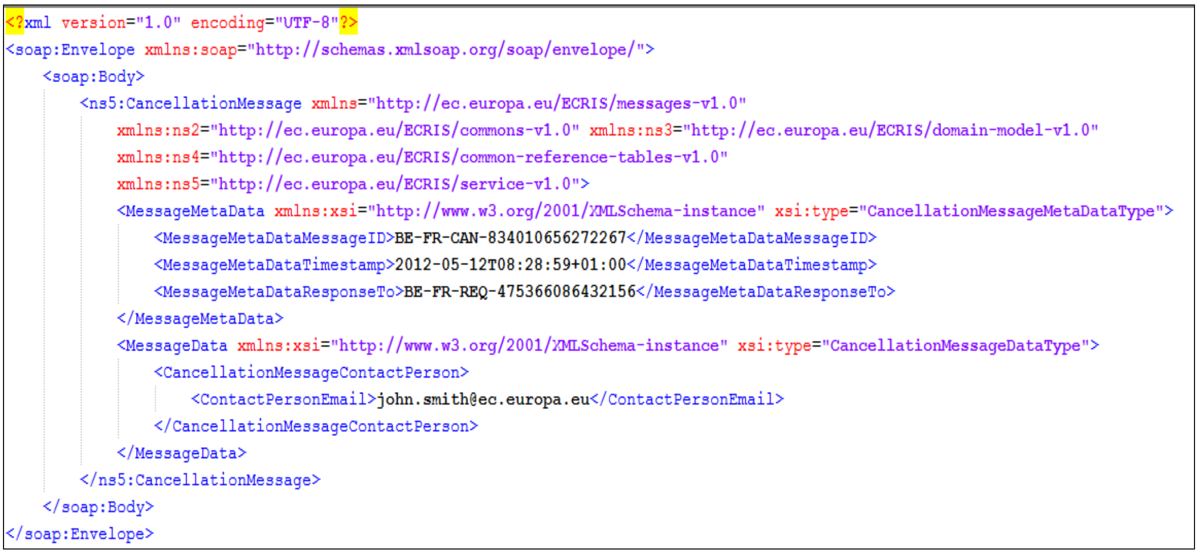
```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:IsAliveMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                            xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                            xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                            xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-IAL-227324156399984</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:28:58+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="IsAliveMessageDataType">
                <IsAlive>true</IsAlive>
            </MessageData>
        </ns5:IsAliveMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 27 – "isAlive" sample message

deliverCancellation

The "deliverCancellation" is the operation that allows cancelling a previously sent message of type request, request response, notification or notification response.

The following figure provides an example of the *SOAP* message:



```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:CancellationMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
            xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0" xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
            xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CancellationMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-CAN-834010656272267</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:28:59+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>BE-FR-REQ-475366086432156</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CancellationMessageDataType">
                <CancellationMessageContactPerson>
                    <ContactPersonEmail>john.smith@ec.europa.eu</ContactPersonEmail>
                </CancellationMessageContactPerson>
            </MessageData>
        </ns5:CancellationMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 28 – "deliverCancellation" sample message

4.4 Handling of Errors

Apart from the business errors which are dealt with by appropriate messages, two types of error responses are supported by the ECRIS technical specifications:

- Synchronous *SOAP* fault messages in response to technical errors.
- Asynchronous functional error messages in response to functional errors.

4.4.1 Technical Errors

As explained previously, although the *web service* operations are defined as "one-way" operations and do not explicitly declare faults as possible returns, the ECRIS software implementation **must throw synchronously a *SOAP* fault** when encountering a low-level technical error (i.e. connectivity errors, XML message does not validate against the XSD, etc.).

Please note that the development platforms that have been tested and that are described later in this document are capable of properly throwing *SOAP* faults when an exception occurs, even if the *SOAP* fault is not explicitly declared in the *WSDL* definition. In addition to this, and because the structural validation of the XML message is performed synchronously when receiving the XML stream, the *web service* implementation automatically throws an adequate technical error if the XML does not respect the *XSD* definitions. In such a case, the error message normally explicitly and clearly indicates which XML element is problematic and why. In such a case, the stack-trace indicates clearly which XML element is problematic and why. It must also be noted that the structure of the standard SOAP faults allow providing a custom error code, a stack-trace and additional custom information that the implementer wishes to log.

4.4.2 Functional Errors

As defined in the "Technical Architecture" document, once the technical transmission of a message has been performed successfully, a **functional validation** of the content of the message can take place **in a second step** and may result in a functional error to be returned

asynchronously to the sender of the message if a problem is encountered. The detailed list of cases that are to provoke such functional errors is provided later in this document. For each such error condition, an appropriate functional error code is defined in the common reference table "FunctionalErrors".

When a functional error occurs, the message to which it relates is to be rejected. Such functional errors are defined for trapping cases of severe inconsistency of the information being transmitted, so as to capture sufficiently early bugs in the software or obvious end-user mistakes. It is expected that when such a functional error occurs, the situation can easily be corrected by the sender who can simply resend the corrected faulty message.

Please note that the functional error message can contain several error codes. Indeed, it is not the aim of sending multiple functional error messages if several problems are detected in one message. If this occurs, only one functional message is to be sent with multiple error codes.

In terms of the **processing** of the functional errors, the following rules are defined:

- A functional error can be sent at any point during a transaction as a response to any message received, but **not after the end** of the transaction or before the start of the transaction. In essence, once a message is received that closes a transaction (i.e. a request response or notification response message), the receiver cannot reply with a functional error message.
- The previous rule implies that if there is a problem in a request response or notification response, it needs to be handled manually by the central authorities outside of ECRIS by human operators. To that end, the ECRIS software on the receiver's end should raise an alert to the staff of the central authority so that the issue can be solved. Please note though that the message cancellation described later in this document facilitates the resolution of such a situation. Indeed, the operator on the receiver's end can inform by any communication means available the sender of the problem that has been encountered. The sender then simply cancels the problematic request/notification response and sends a corrected message using the ECRIS software.
- A functional error message ends the transaction in the following cases:
  - when issued as a response to a notification message
  - when issued as a response to a request message, independently of whether the "request deadline" message has been sent before or not; concretely, this means that both sequences "request → functional error" and "request → request deadline → functional error" lead to the end of the transaction
- A functional error message cannot be answered with another functional error (this avoids endless loops).
- Given the previous rule, a message following a functional error must be correlated with the last message received in the transaction and that was not a functional error. For example, Member State "A" sends a request to Member State "B". Member State "B" sends back the request deadline. Then "B" delivers a request for additional information to "A". Let's assume that the response of "A" to the request for additional information of "B" is problematic and triggers a functional error. "A" receives the functional error and re-issues a new corrected response to the request for additional information. This response message must be correlated with the request for additional information message issued by "B".
- Considering the rules above, the kinematics of ECRIS could potentially generate an infinite loop if the sending of additional identification information systematically raises a functional error. Concretely, if the ECRIS processing is fully automated, the following could occur: request → request deadline → request for additional ID info → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → …

  In order to avoid this situation, in the case that the "additional ID info" message raises the same functional error 3 times consecutively within one transaction, the 3rd time the receiving ECRIS software **must drop** the "additional ID info" message without replying to it and consider that no additional information is available. This allows the receiving

Member State to exit the loop and to continue the request or notification process so as to read the proper end of the transaction.

4.4.3 Message Cancellation

As described in the list of *web service* operations, the ECRIS detailed technical specifications include a specific cancellation message named "deliverCancellation" that allows annulling a previously sent message so as to be able to correct mistakes.

Please note first that "cancellation" is defined in this context as the **annulment** of a previously transmitted ECRIS **message** because it was sent wrongfully or because it contains erroneous information (i.e. as a result of a bug in the system or of a wrong manipulation by an end-user of the system). It must not be confused with the judicial means for cancelling convictions. This is handled through the rehabilitation mechanism. The domain model already foresees the necessary elements for informing the Member State of nationality that a conviction has been rehabilitated and that it is to be removed from the criminal records register. The "cancellation" is to be understood here on the technical level and as a functionality allowing correcting operational or technical mistakes.

The following rules apply for the usage of this "deliverCancellation" operation:

- **Only** the following types of messages can be cancelled:
  - Request
  - Request response
  - Notification
  - Notification response
- Trying to cancel a message which does not belong to the list above results in a functional error (such a functional error is actually the only direct response that can be provided to a cancellation message).
- The cancellation message itself can **never** be cancelled by a subsequent cancellation message.
- **Only** the sender of a message is allowed to cancel that message.
- Cancelling a request response or a notification response message in a given transaction does not stop the transaction itself while cancelling a request or a notification message stops the entire transaction.

  Thus, in the situation where a request or notification message is cancelled by the requesting or notifying Member State, the transaction is considered as closed and therefore no other messages are expected to be exchanged (e.g. no notification response, request response, request deadline or request for additional information). If such messages are still received as a response to a message that has been cancelled, they are to be handled as unexpected messages, as described later in chapter §4.6.1.
- In the case where a notification response or a request response message is being cancelled, the transaction is reopened. Please note that the **initial deadline** for the transaction **is still valid and is not reset** by cancelling the notification/request response. In such a case, a new message of the same type as the cancelled one, carrying the corrected set of information, is expected to be sent so as to close again the transaction within its deadline. The possibility to cancel such messages is however further limited in time:
  - A request response can be cancelled within one week (i.e. 7 days) after it was sent, provided that the initial transaction deadline has not elapsed.
  - A notification response can be cancelled within one month (i.e. 30 days) after it was sent, provided that the initial transaction deadline has not elapsed.

  For example, if a "request response" message is cancelled, the expected behaviour is to resend a new "request response" message for correcting the cancelled one. When re-sending a corrected message it must however be noted that the technical message ID of the new message must be changed so as to remain unique across all ECRIS messages. This message cannot reuse the unique technical identifier of the cancelled message.

Please note also that if the notification/request response message is cancelled at a moment in time that is very close to the end of the deadline of the notification/request transaction, there may be no sufficient time for actually resending the corrected notification/request response message. In such a borderline case, the central authority having cancelled the problematic message must take contact with the corresponding Member State's central authority in order to solve the issue manually.

- Given the previous rules, a corrected request response or notification response message that is sent after having been cancelled must correlate with the last message received in the transaction and that was not a functional error or the cancellation message itself. For example, Member State "A" sends a request to Member State "B". Member State "B" sends back the request deadline and then the request response. In the central authority of "B", an operator realises that the request response has a severe mistake. "B" sends a cancellation message to "A" for annulling the request response that was just sent. When "B" sends the corrected request response message to "A", it must correlate with the request message issued by "A" since this is the last business message received by "B" in this transaction.
- The previous rules also imply that if a problem is detected for a request response or notification response message after the cancellation time limits defined above, the central authority having sent the problematic message can no longer cancel it via ECRIS but must take contact with the corresponding Member State's central authority in order to solve the issue manually.

Please note that the cancellation mechanism is meant to be used for correcting significant mistakes, such as providing the wrong identity of a person or the wrong list of convictions. Minor corrections to notification messages can be easily transmitted by simply resending the corrected notification message and indicating that this new conviction information updates conviction information previously sent.

Since a cancellation message is **not to be used** for indicating a judicial removal of a conviction but only to correct erroneous situations that occur by mistake, the receiver of such a cancellation message knows for sure that the message that the sender wants to cancel must be dropped and that the information contained in that message must simply be ignored or deleted.

4.5 Handling of Binary Attachments

The ECRIS detailed technical specifications include the support for exchanging fingerprints in the form of binary attachments to the XML messages. These binary attachments are optional and may only carry NIST files, as defined in the "Technical Architecture" and "Business Analysis" documents.

More specifically, the "Technical Architecture" document states:

"As a result, the optional exchange of fingerprints in ECRIS is to be performed by joining NIST files as binary attachments to the *XML* messages to be exchanged between the Member States. The NIST file should primarily contain the ten-print fingerprint image and optionally the palm-print images (if available), as grey-scale images of a resolution of 500 dpi, encoded and compressed with the "Wavelet Scalar Quantization" algorithm (WSQ).

Please note that the definition of the detailed content of the NIST file is out of scope of the *ECRIS Technical Specifications* project. It is therefore recommended to apply the same standard for NIST files as the one that has been defined for the PRÜM project. The detailed definition of this standard can be found in the Council Decision 2008/616/JHA of 23 June 2008 on the implementation of Decision 2008/615/JHA on the stepping up of cross-border cooperation, particularly in combating terrorism and cross-border crime, more specifically in "CHAPTER 2: Exchange of dactyloscopic data (interface control document)" of the annex."

The support for such binary attachments is implemented as follows in the technical artefacts:

- The type "BinaryAttachmentType" contains only the meta-data of a binary attachment file; the type "BinaryAttachmentPushType" is an extension of the previous type and contains additionally the binary file itself.

- "Pull" implementation:

  All regular messages that can carry identification information can optionally contain an element of type "BinaryAttachmentType", indicating that fingerprints are available. If a Member State receives such a message, it can use the "deliverRequestAdditionalInfo" operation, using the "BinaryAttachmentID" value provided from the received meta-data as parameter, in order to ask the sending Member State to send the binary file. The Member State having the electronic fingerprints transmits the binary file using the "deliverRequestAdditionalInfoResponsePush" operation, which contains an element of type "BinaryAttachmentPushType".

- "Push" implementation:

  The "push" versions of the messages can optionally contain an element of type "BinaryAttachmentPushType" which immediately carries the fingerprints binary file. These operations are however only to be used if supported by both the sending and the receiving Member State. An implementation that does not support the usage of the "push" operations must systematically throw a *SOAP* fault using the specific fault code "OperationNotSupported".

Please note that the service contract itself cannot enforce the rules relating to the type and size of binary files that are defined in the "Technical Architecture" document, due to the fact that specifications such as *Xmime* or *MTOM-Policy* were discarded for interoperability reasons. These controls are to be implemented in the ECRIS software and appropriate functional error messages are to be returned if an error situation occurs.

The following sequence diagram illustrates the usage of the "pull" services for pulling a binary file:

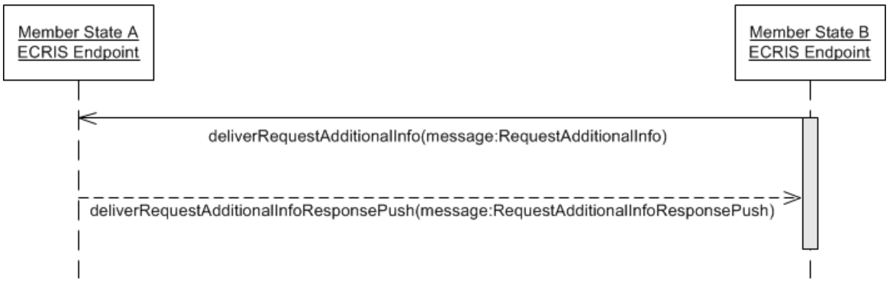

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequestAdditionalInfo(message:RequestAdditionalInfo)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePush)

Figure 29 – Pull implementation kinematic

This sequence can only occur within an on-going request of notification transaction and if the sender of the request or notification message provided information about available fingerprints. In this case, the receiver of the message can ask the sender to provide the NIST file by using the "request additional info" message and asking specifically in this message to retrieve the binary attachment by providing the ID of the binary attachment (i.e. field "MessageDataBinaryIDRequested") and using the value "fingerprints" in the field "MessageDataAdditionalInformationRequested". The sender then pushes the electronic file by using the "push" version of the "response to request for additional info" message.

The following diagram illustrates the pulling of the binary attachment in the context of a request kinematic:

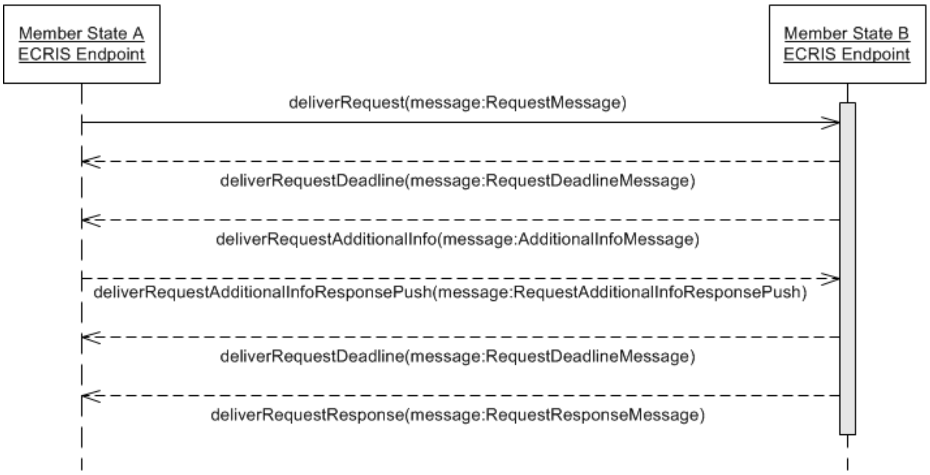

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePush)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 30 – Request kinematic using the pull implementation

4.6 Kinematics

The following chapters illustrate the expected kinematics supported by the ECRIS technical specifications, using UML sequence diagrams. The kinematics described in the following sections cover **all** message exchanges to be supported for ECRIS. They are grouped based on the business process that is supported (i.e. notification or request).

Please note that the more general processing rules for functional errors and cancellation of messages defined in this document must always be kept under consideration and apply to the kinematics presented. Keeping this in mind, the sequence diagrams do not depict all possible combinations of notifications and requests together with functional errors or cancellation messages for the sake of readability.

For the same reason, version-related information such as suffixes in operations or messages has been omitted in the diagrams. The following examples also assume that no technical error occurs resulting in a synchronous SOAP fault to be returned.

4.6.1 Handling Unexpected Messages

As a general rule, an ECRIS message is to be considered **unexpected** if it is not foreseen in any of the kinematics described further in this document. Functional error and cancellation messages are also to be considered **unexpected** in case that the rules defined for handling and sending them are not respected (for example a functional error message being sent as an answer to a request response message).

Depending on the context in which an unexpected message is received, an ECRIS implementation must follow the rules below:

- A message is received outside of any on-going transaction. As an example: Member State "A" has currently no on-going "request" transactions open with Member State "B" but receives from "B" a "request deadline" message. In such a case, the receiver of the unexpected message (1) drops the message, (2) does not reply to the sender automatically and (3) raises an alert to an operator. Indeed, such messages could originate from a security breach and be an attempt of an outside threat to impersonate a Member State.
- A message is received within an on-going transaction but does not correlate to another message previously exchanged in this transaction. In particular, it refers to a message ID (1) that is not known to the receiver or (2) that belongs to another transaction or (3) that does not relate to the previous message but to another one in the transaction. In such cases, the receiver should send back a functional error with the appropriate error code to the sender (i.e. functional error code GEN-2).
- A message is received within an on-going transaction and it does correlate to the previous message but it is not supposed to occur at this moment in the transaction.

Example: request → request deadline → ask for additional ID information → request deadline; in this case the last "request deadline" message is unexpected because no response has been provided yet to the request for additional ID information.

In such cases, the receiver should send back a functional error with the appropriate error code to the sender (i.e. functional error code GEN-3).

4.6.2 Notifications

This chapter illustrates the kinematics supported for the message exchanges in the context of "Notify Convictions and Subsequent Changes" business process. The kinematics described in the following sections cover **all possible and expected combinations** for notifications.

However, as mentioned earlier, a functional error can occur within this transaction at several points as a response to a message. Given that, not all possible functional error combinations are depicted but only a few sample cases are provided.

Deliver notification

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B which processes the notification and sends a notification response back.

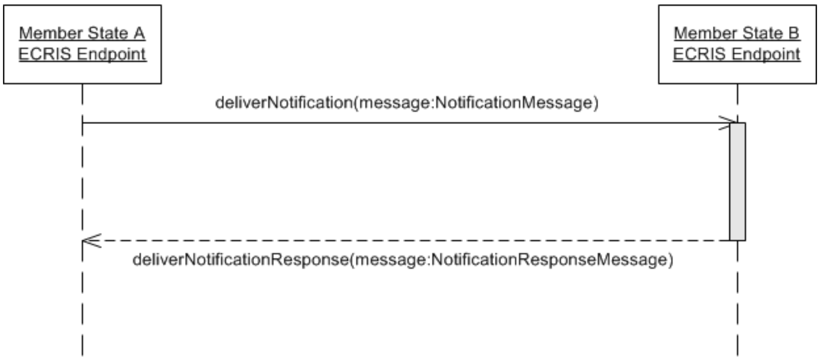

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 31 – "Deliver notification" kinematic

Please note that this kinematic does not support only successful notification message exchanges but also exchanges that result in a business error message. The exact outcome of the processing of the notification is provided by the "MessageDataNotificationResponseTypeReference" property of the "NotificationResponseMessage" element.

In any case, the notification response closes the transaction.

Deliver notification («Push» version)

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. All other mechanics and rules already described earlier remain the same.

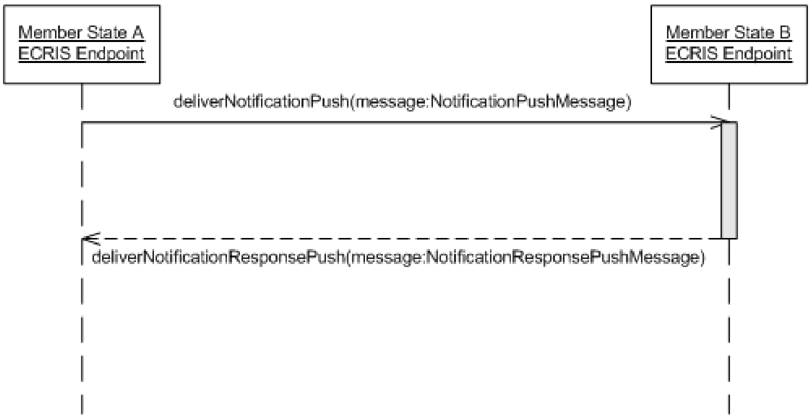

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotificationPush(message:NotificationPushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 32 – "Deliver notification ("push" version) kinematic

Deliver notification > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B which encounters a functional error when processing the notification. Member State B sends a functional error message back to Member State A.

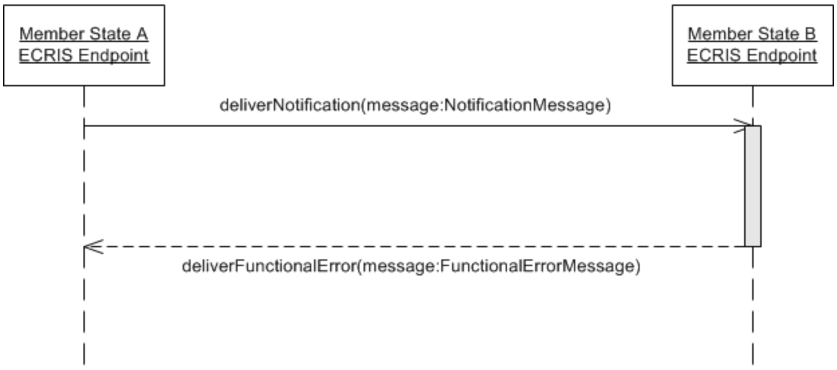

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverFunctionalError(message:FunctionalErrorMessage)

Figure 33 – "Deliver notification > functional error" kinematic

Please note that, as already explained earlier, in this scenario the functional error response also closes the transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the notification message (i.e. "deliverNotificationPush" operation).

Deliver notification > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. Shortly afterwards, an operator within the central authority of Member State A realises a significant problem in the information transmitted. Therefore, he decides to issue a cancellation message for annulling the notification already sent.

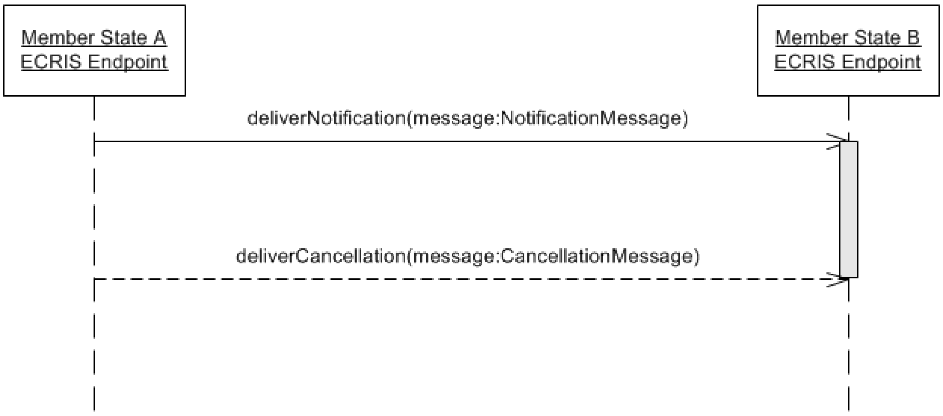

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverCancellation(message:CancellationMessage)

Figure 34 – "Deliver notification > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message also closes the transaction. The operator of Member State A can then correct the problem and if required send the corrected notification later, which will initialise a new transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the notification message (i.e. "deliverNotificationPush" operation).

Deliver notification > notification response > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. Member State B sends the notification response but shortly afterwards, an operator within the central authority of Member State B realises a significant problem in the response transmitted. Therefore, he decides to issue a cancellation message for annulling the notification response already sent.

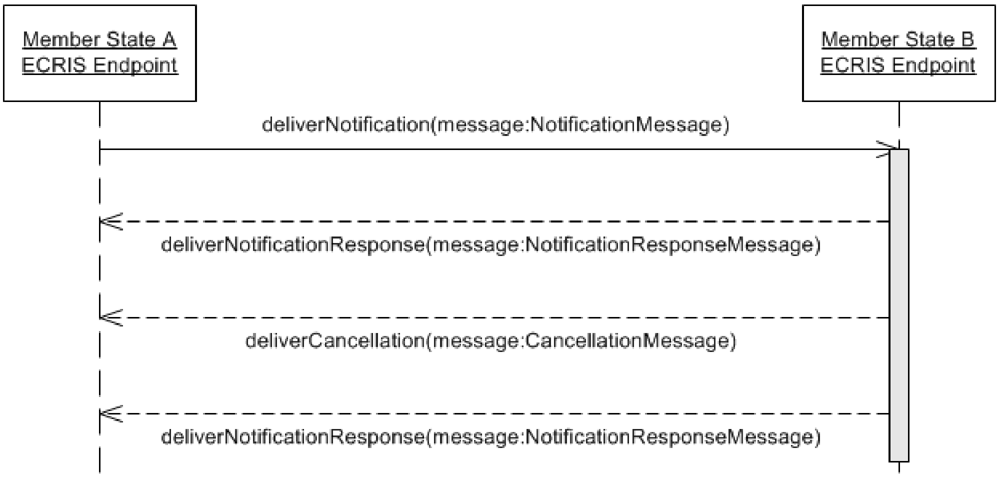

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)
deliverCancellation(message:CancellationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 35 – "Deliver notification > notification response > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message reopens the previously closed transaction. The central authority of Member State B is expected to correct the problem and to send a corrected notification response later, which will again close the transaction.

Please note also that this kinematic is identical for the "push" version of the notification messages (i.e. "deliverNotificationPush" and "deliverNotificationResponsePush" operations).

Deliver notification > request additional ID info

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B sends a notification response back to Member State A.

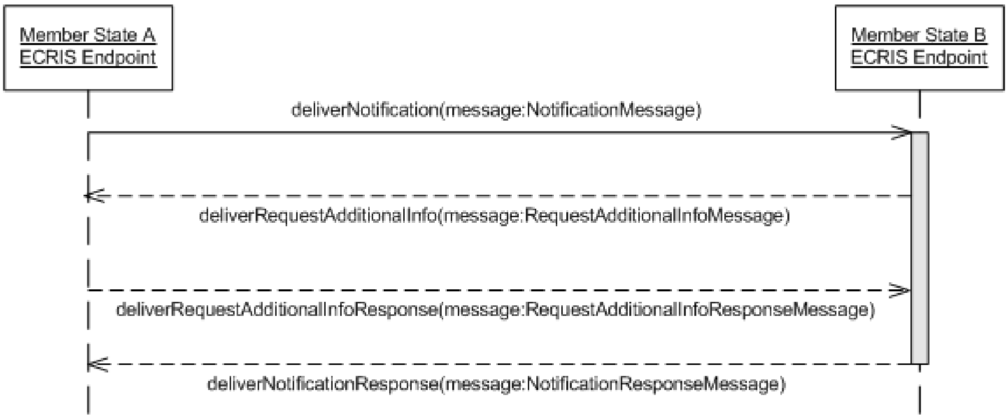

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverRequestAdditionalInfo(message:RequestAdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:RequestAdditionalInfoResponseMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 36 – "Deliver notification > request additional ID info" kinematic

Please note that this kinematic does not support only the case where additional information is available but also the case where such information is unavailable. The exact outcome of the processing of the request for additional identification information is provided by the "MessageDataAdditionalInfoResponseTypeReference" property of the "RequestAdditionalInfoResponseMessage" element.

Also, a Member State receiving a notification can, in theory, issue requests for additional information multiple times during the transaction. This case is very similar to the "deliver notification > request additional info" kinematic but is not further elaborated in this document. It should however be supported by the ECRIS implementations, following the principle described here.

In any case, the notification response closes the transaction.

Please note also that the fact that additional identification information is received does not have any impact on the notification's deadline (i.e. the operational deadline as defined in the "Business Analysis" document). In particular, no new deadline is calculated and sent by the notified Member State to the notifying Member Sate.

Deliver notification («Push» version) > request additional ID info

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics and rules already described earlier remain the same.

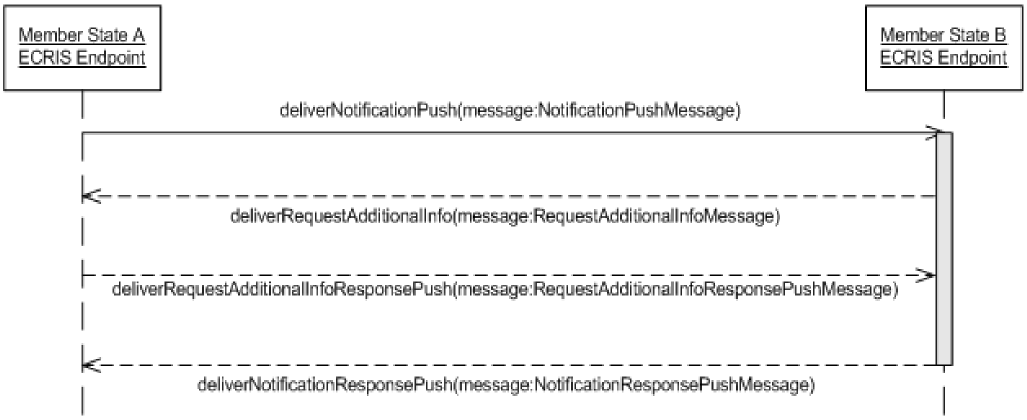

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotificationPush(message:NotificationPushMessage)
deliverRequestAdditionalInfo(message:RequestAdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 37 – "Deliver notification ("push" version) > request additional ID info" kinematic

Deliver notification > request additional ID info > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B encounters a functional error when processing the additional identification information transmitted by Member State A. Member State B sends a functional error message back to Member State A. Member State A sends a corrected version of the additional identification information to Member State B. Finally Member State B sends a notification response back to Member State A.

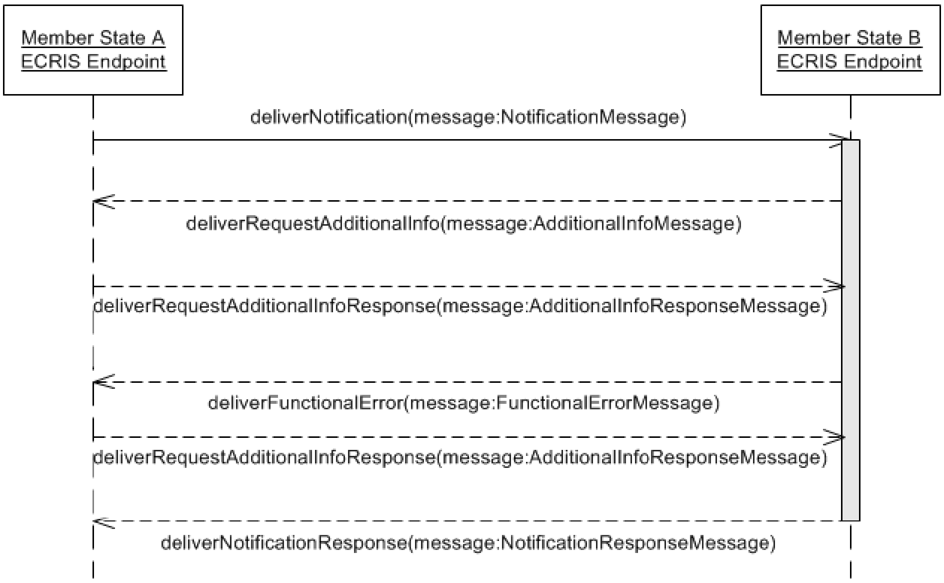

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 38 – "Deliver notification > request additional ID info > functional error" kinematic

Please note that in this case the functional error message does **not** close the transaction as the Member State providing additional information can make the appropriate corrections and issue again the corrected information. Please also note that the "deliverFunctionalError" operation can be invoked at any point **after** the "deliverNotification" operation and **before** the "deliverNotificationResponse" operation are invoked.

Deliver notification («Push» version) > request additional ID info > functional error occurs

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics and rules already described earlier remain the same.

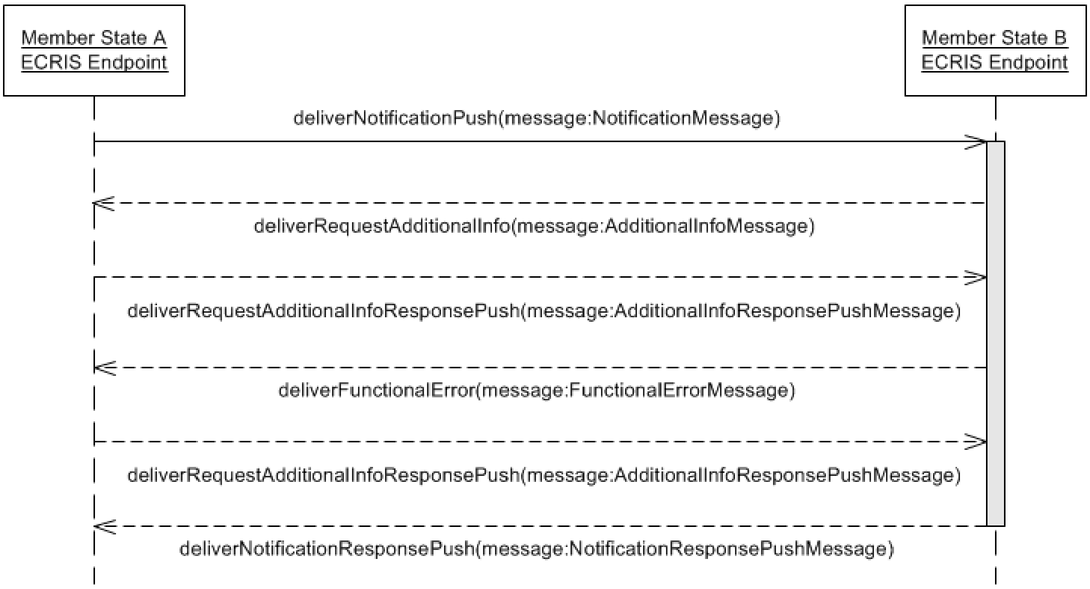

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotificationPush(message:NotificationMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 39 – "Deliver notification ("push" version) > request additional ID info > functional error" kinematic

4.6.3 Requests

This chapter illustrates the kinematics supported for the message exchanges in the context of the "Request Criminal Record Information" business process. The kinematics described in the following sections cover **all possible and expected combinations** for requests.

However, as mentioned earlier, a functional error can occur within this transaction at several points as a response to a message. Given that, not all possible functional error combinations are depicted but only a few sample cases are provided.

Deliver request

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline to Member State A, processes the request and sends a request response back.

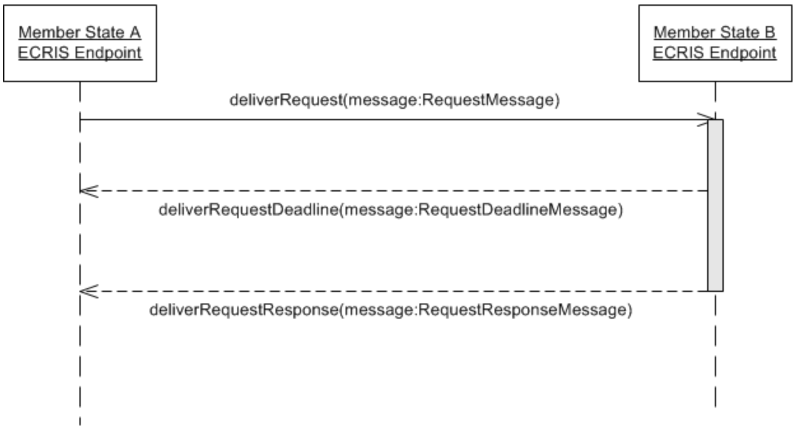

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 40 – "Deliver request" kinematic

Please note that this kinematic does not support only successful request message exchanges but also exchanges that result in a business error message during the processing or in a request denial. The exact outcome of the processing of the request is provided by the "MessageDataRequestResponseTypeReference" property of the "RequestResponseMessage" element.

In any case, the request response closes the transaction.

Deliver request («Push» version)

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. All other mechanics and rules already described earlier remain the same.

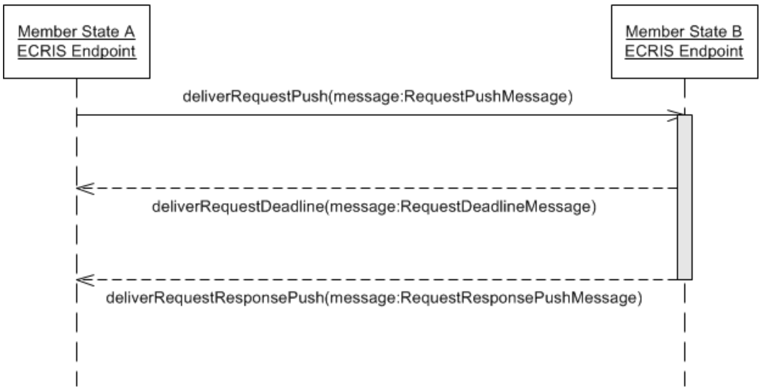

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequestPush(message:RequestPushMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponsePush(message:RequestResponsePushMessage)

Figure 41 – "Deliver request ("push" version)" kinematic

Deliver request > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which encounters a functional error when processing the request message. Member State B sends a functional error message back to Member State A.

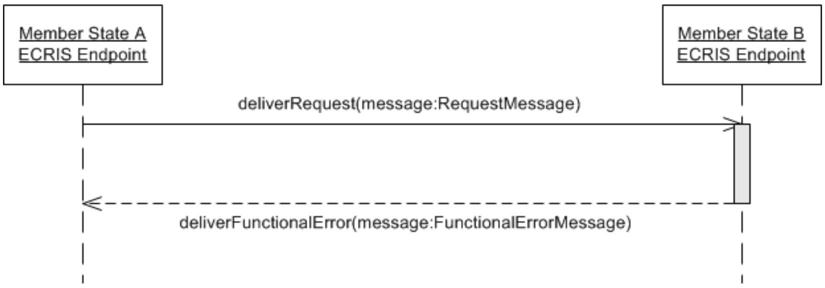

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverFunctionalError(message:FunctionalErrorMessage)

Figure 42 – "Deliver request > functional error" kinematic

Please note that, as already explained earlier, in this scenario the functional error response also closes the transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the request message (i.e. "deliverRequestPush" operation).

Deliver request > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B. Member State B receives the request and sends to Member State A the deadline for replying the request. At some point, before the deadline is expired or a reply is issued, the central authority in Member State A that authored the request realises that a mistake was made in this request. At that point, it issues a cancellation message for annulling the request.

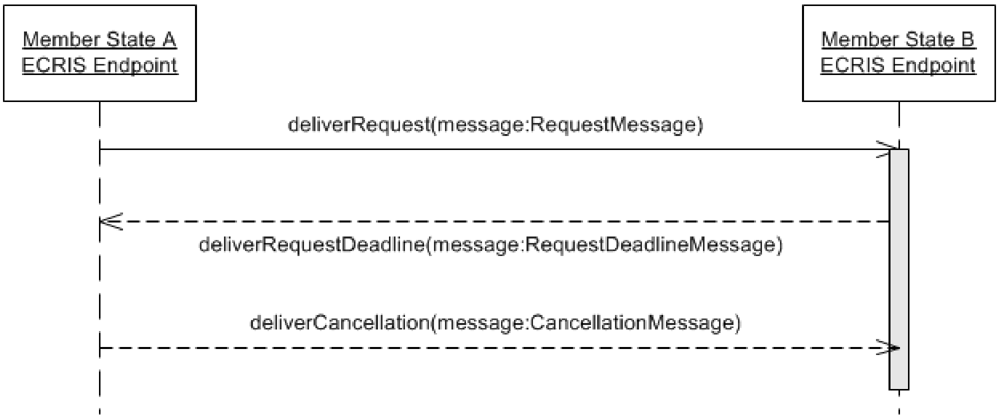

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverCancellation(message:CancellationMessage)

Figure 43 – "Deliver request > cancellation" kinematic

As mentioned earlier, the cancellation message for a request essentially terminates the message exchange. Thus, Member State B has to consider the transaction closed and stop any further actions so as to satisfy the request.

Please note that it is recommended to fully automate the calculation and sending of the legal deadline for requests. Therefore the diagram above illustrates the expected sequence of operations. However, the cancellation can also theoretically occur before sending the deadline for the request and must also be supported as a valid kinematic, as depicted below:

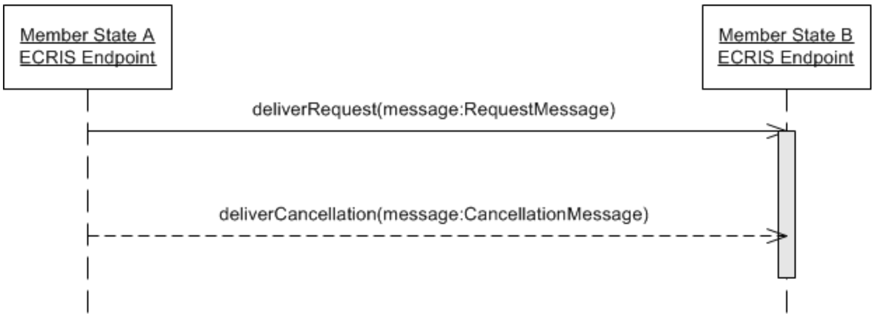

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverCancellation(message:CancellationMessage)

Figure 44 – "Deliver request > cancellation" kinematic

Deliver request > request response > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B. Member State B transmits the legal deadline and the response to the request but shortly afterwards, an operator within the central authority of Member State B realises a significant problem in the response transmitted. Therefore, he decides to issue a cancellation message for annulling the request response already sent.

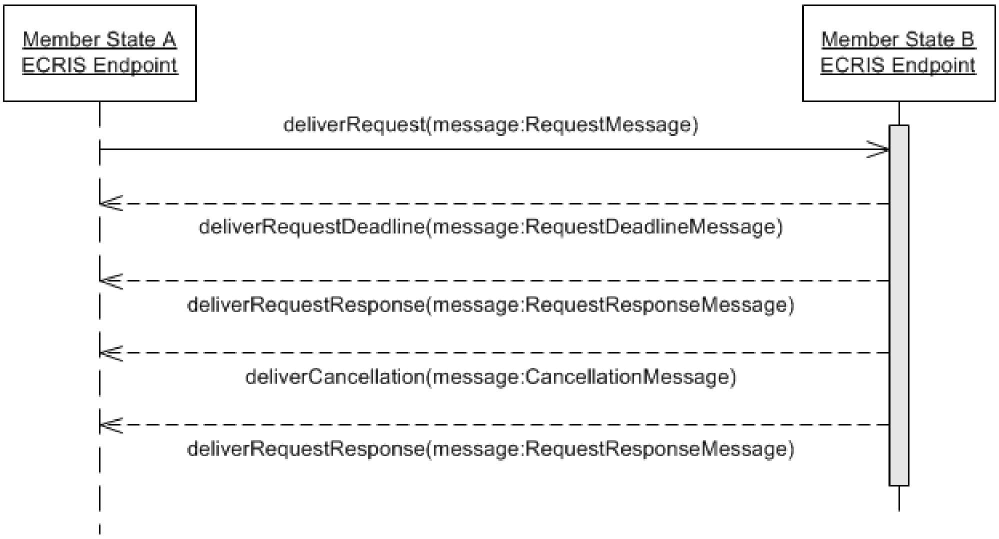

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)
deliverCancellation(message:CancellationMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 45 – "Deliver request > request response > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message reopens the previously closed transaction. The central authority of Member State B is expected to correct the problem and to send a corrected request response later, which will again close the transaction.

This kinematic is identical for the "push" version of the request messages (i.e. "deliverRequestPush" and "deliverRequestResponsePush" operations).

Deliver request > request additional ID info

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline back to Member State A. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B sends a request response back to Member State A.

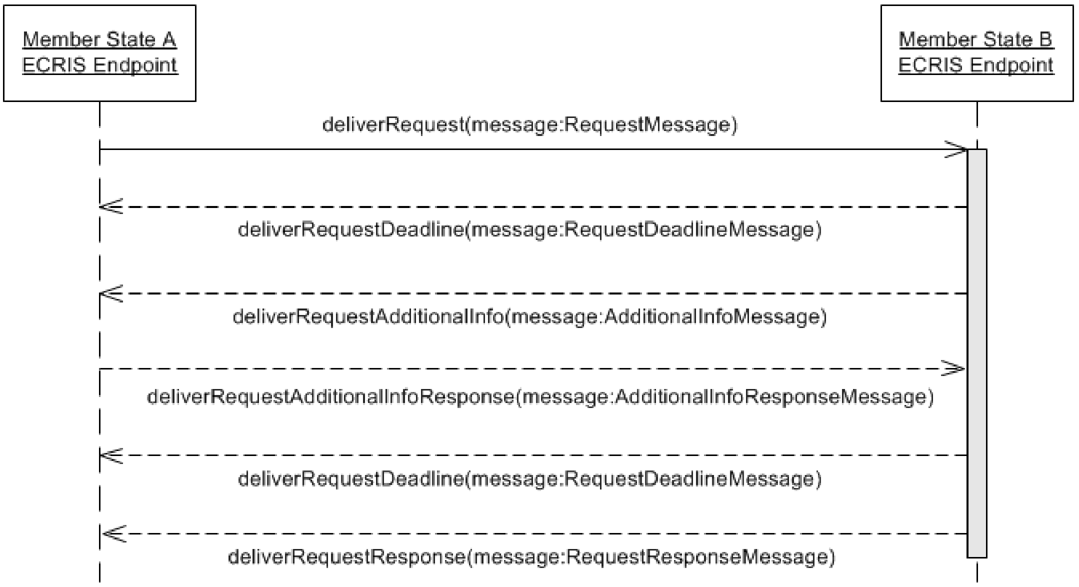

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 46 – "Deliver request > request additional ID info" kinematic

Please note that this kinematic does not support only the case where additional information is available but also the case where such information is unavailable. The exact outcome of the processing of the request for additional identification information is provided by the "MessageDataAdditionalInfoResponseTypeReference" property of the "RequestAdditionalInfoResponseMessage" element.

Also, a Member State receiving a request can, in theory, issue requests for additional information multiple times during the transaction. This case is very similar to the "deliver request > request additional info" kinematic but is not further elaborated in this document. It should however be supported by the ECRIS implementations, following the principle described here.

Here also, the diagram above illustrates the expected sequence of operations. However, the request for additional information can also theoretically occur before sending the deadline for the request and must also be supported as a valid kinematic, as depicted below:

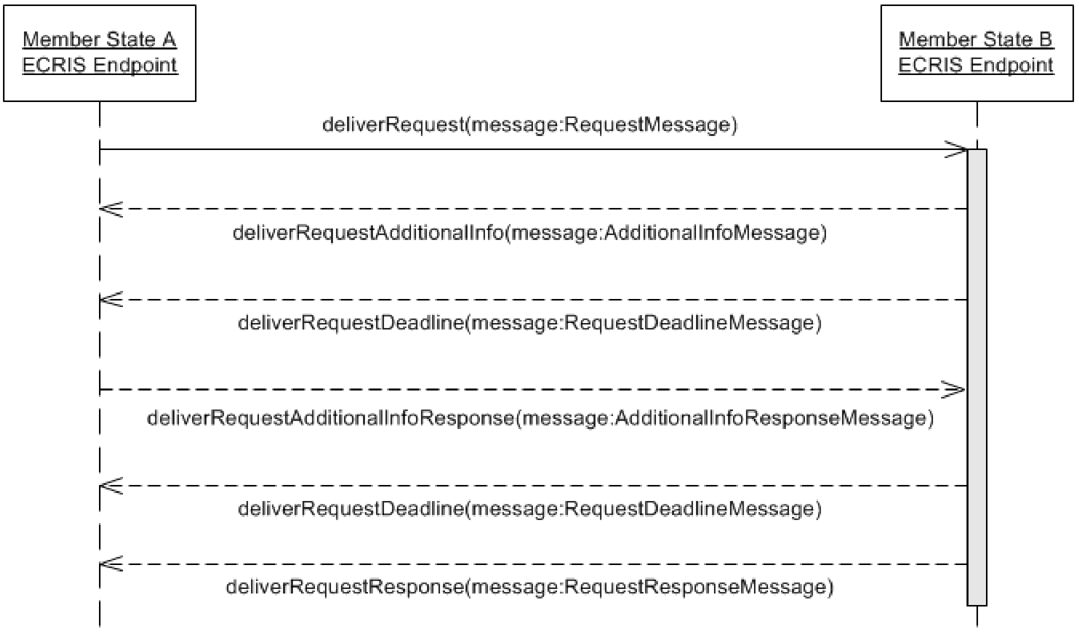

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 47 – "Deliver request > request additional ID info" kinematic

In any case, the request response closes the transaction.

Deliver request («Push» version) > request additional ID info

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics, rules and variations already described earlier remain the same. In particular here also the request for additional information can theoretically occur before sending the deadline for the request.

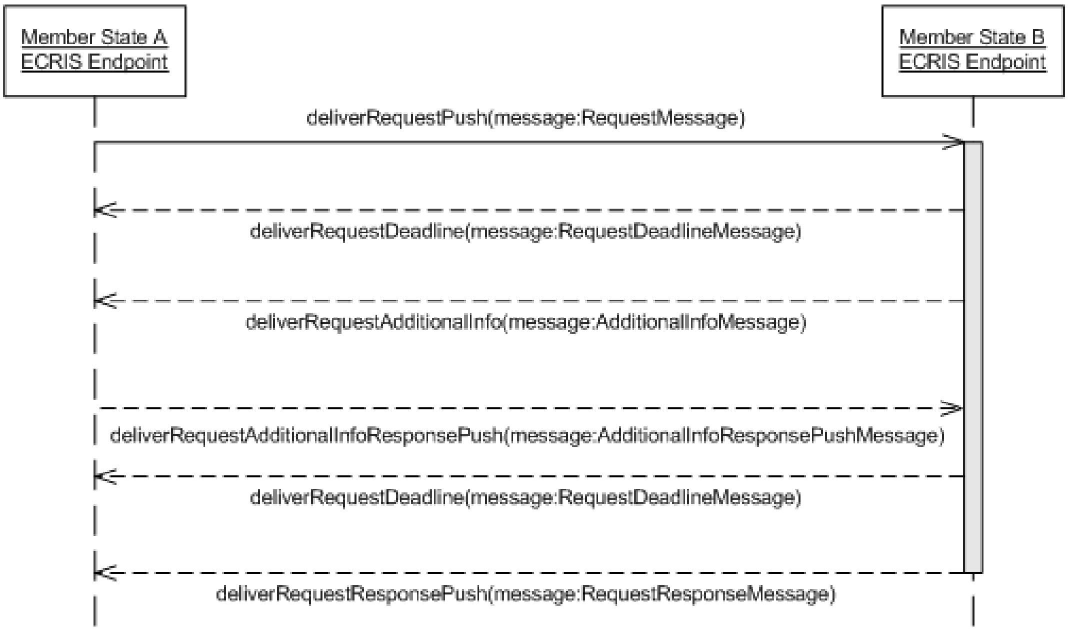

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequestPush(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponsePush(message:RequestResponseMessage)

Figure 48 – "Deliver request ("push" version) > request additional ID info" kinematic

Deliver request > request additional ID info > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline back to Member State A. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B encounters a functional error when processing the additional identification information transmitted by Member State A. Member State B sends a functional error message back to Member State A. Member State A sends a corrected version of the additional identification information to Member State B. Finally, Member State B sends a request response back to Member State A.

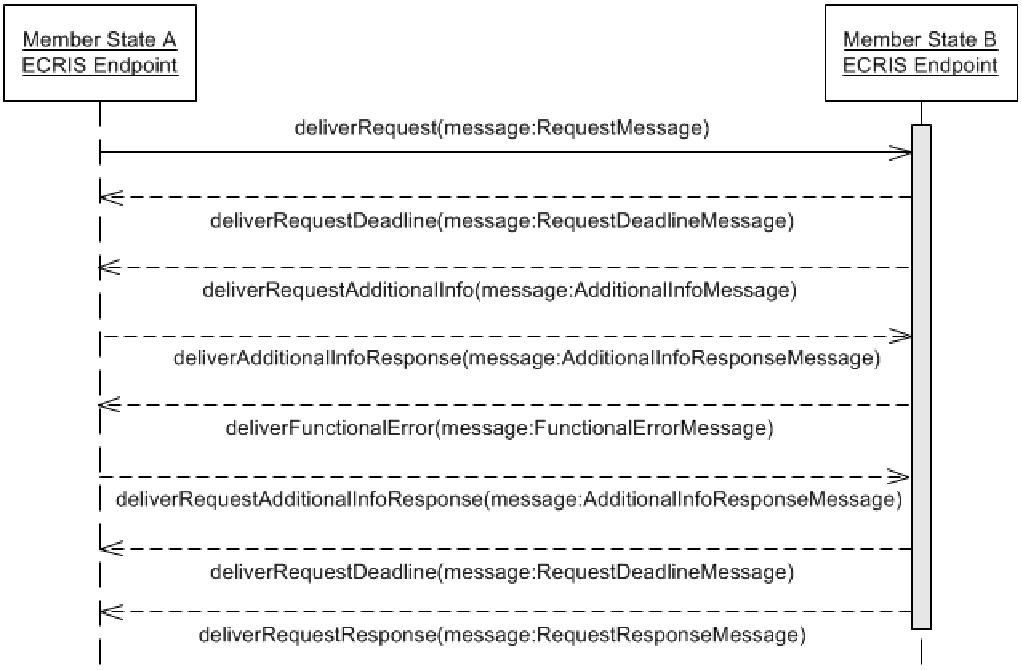

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 49 – "Deliver request > request additional ID info > functional error" kinematic

Please note that in this case the functional error message does **not** close the transaction as the Member State providing additional information can make the appropriate corrections and issue again the corrected information. Please also note that the "deliverFunctionalError" operation can be invoked at any point **after** the "deliverRequest" operation and **before** the "deliverRequestResponse" operation are invoked.

Here also, the diagram above illustrates the expected sequence of operations. However, the sending of the new deadline for the request can theoretically also occur before sending the functional error and must also be supported as a valid kinematic, as depicted below:

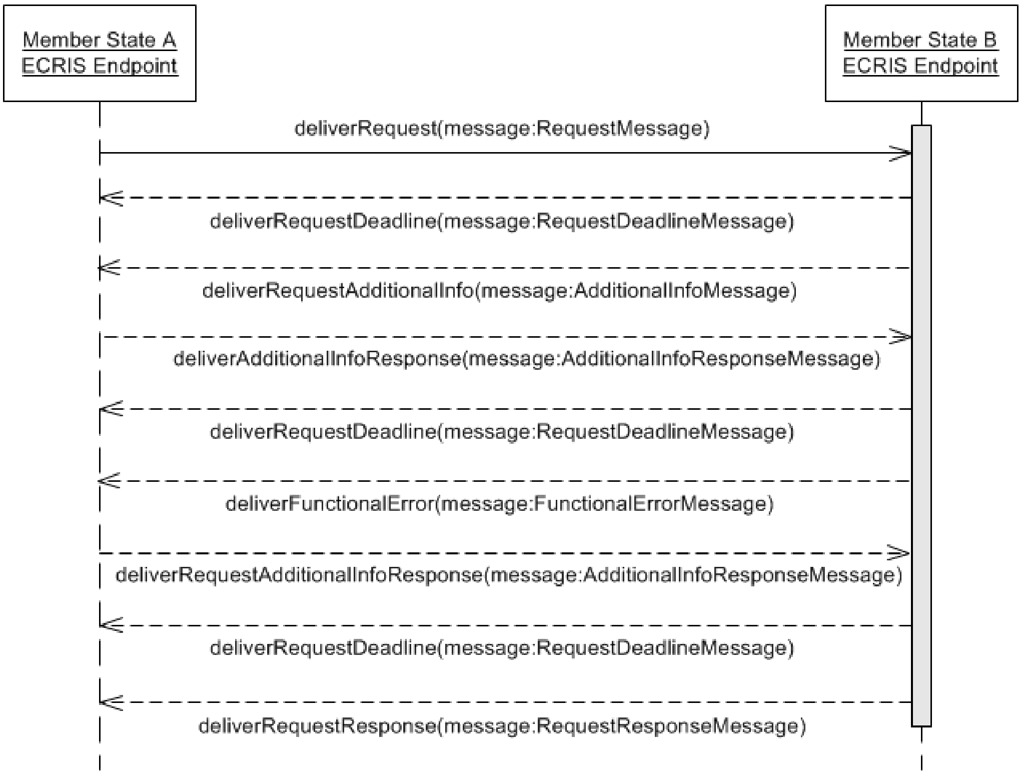

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 50 – "Deliver request > request additional ID info > functional error" kinematic

Given the variations that are also possible for sequence of occurrence of the first deadline message, the following diagrams also illustrate valid kinematics that need to be supported by the ECRIS software:

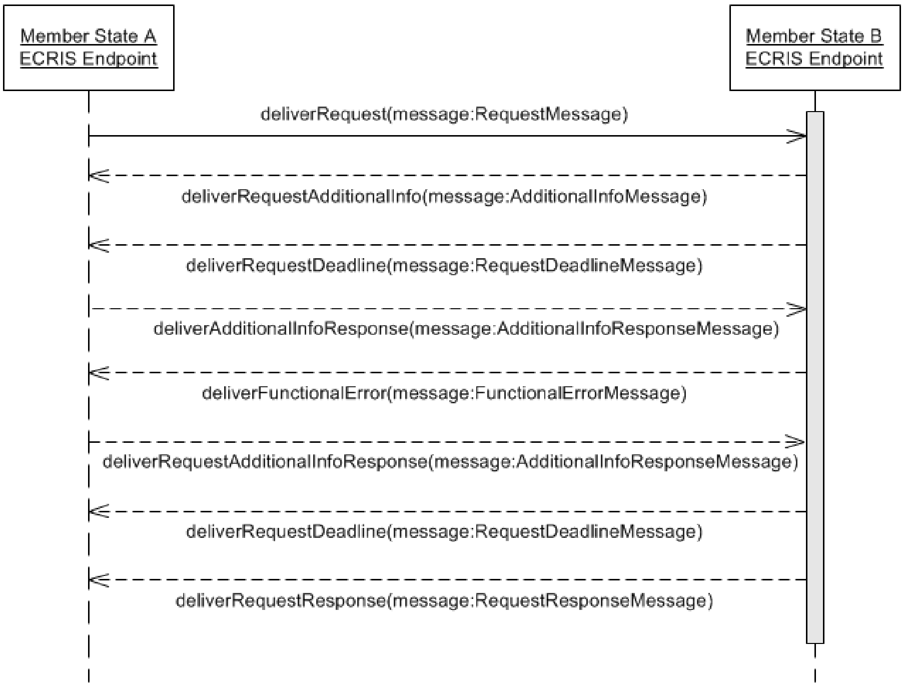

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 51 – "Deliver request > request additional ID info > functional error" kinematic

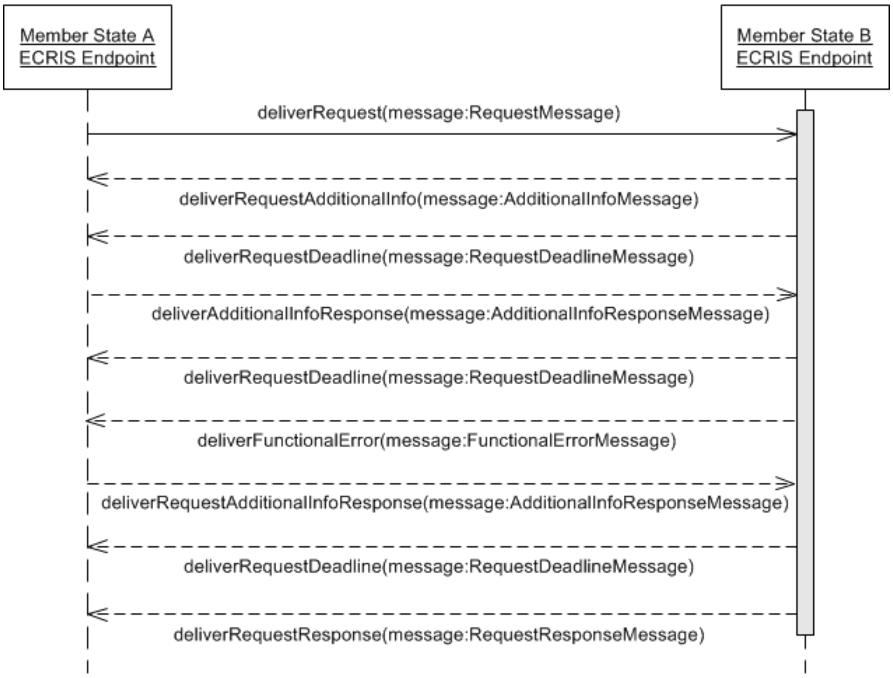

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 52 – "Deliver request > request additional ID info > functional error" kinematic

In any case, the request response closes the transaction.

Deliver request («Push» version) > request additional ID info > functional error occurs

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation. All other variations described in the previous request kinematics also apply to the "push" version of this message exchange.

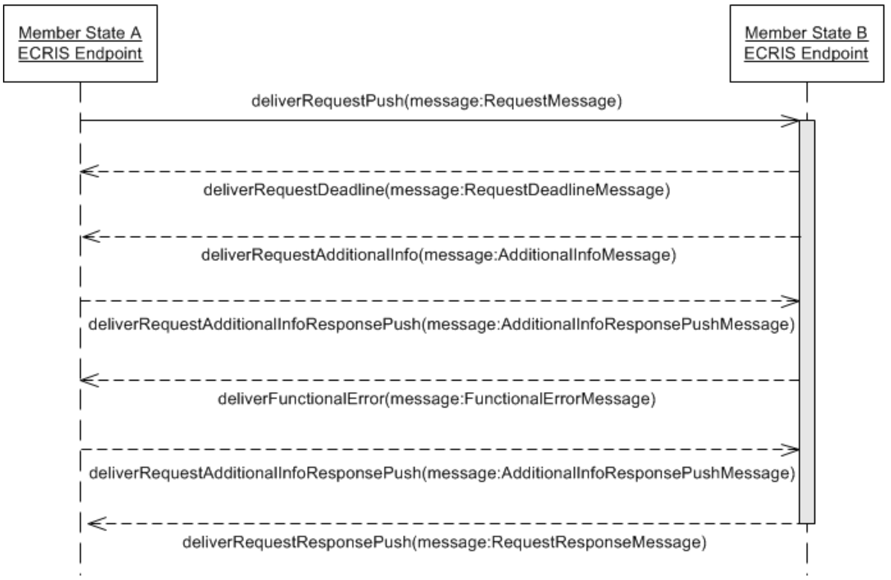

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequestPush(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverRequestResponsePush(message:RequestResponseMessage)

Figure 53 – "Deliver request ("push" version) > request additional ID info > functional error" kinematic

5 IMPLEMENTATION

This chapter provides rules, guidelines and recommendations to the implementers of the detailed technical specifications of ECRIS.

The rules defined below have to be followed **strictly** by all software implementations so as to ensure the proper operation of the software and effective data exchanges between Member States.

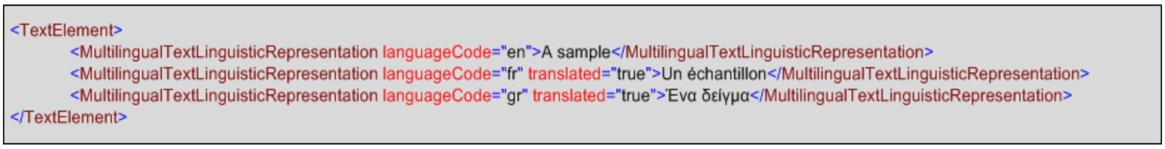
The guidelines and recommendations described further below are based on:

- Practical experience from the NJR pilot project
- Lessons learned gathered during the design and validation of the technical artefacts
- Best practices of software integration

5.1 Transmitting Text in Several Languages

Given its nature, ECRIS needs to support transmission of translated and/or transliterated text. To that end, a specific structure called "MultilingualTextType" is defined in the domain model. This type provides the necessary facilities for sending the same text value in various languages. It also allows indicating whether a text value is the result of translation and/or transliteration.

This functionality is achieved by defining a property with unbounded cardinality for "MultilingualTextType", called "MultilingualTextLinguisticRepresentation". This property can take any text value and offers a set of properties for designating whether the text is translated or transliterated. In the following sample, a "TextElement" of type "MultilingualTextType" is used to transmit the text value "A sample" in three different languages.



```
<TextElement>
        <MultilingualTextLinguisticRepresentation languageCode="en">A sample</MultilingualTextLinguisticRepresentation>
        <MultilingualTextLinguisticRepresentation languageCode="fr" translated="true">Un échantillon</MultilingualTextLinguisticRepresentation>
        <MultilingualTextLinguisticRepresentation languageCode="gr" translated="true">Ένα δείγμα</MultilingualTextLinguisticRepresentation>
</TextElement>
```

Figure 54 – A sample of using "MultilingualTextType"

There are at least two use cases that have been identified and that would require the usage of such multi-lingual text representation:

- By the ECRIS legal basis, request messages have to be sent in the language of the requested Member State. In the "Business Analysis" it has been defined to also be able to send the request additionally in the language of the sender. With the proposed solution, the same request message can now contain free text elements in both languages at the same time.
- Request responses can be constituted of information extracted from the national register to which foreign convictions that have been stored outside of the register for the sole purpose of retransmission are added. In such cases, the response to a request can contain text in multiple languages. Also, for the foreign convictions that were stored in the national register, it may be interesting to be able to send the texts that were translated in both their original language and in the translated language.

5.2 Correlating Convictions

As defined in the "Business Analysis" document, it is required to be able to establish relations with previously transmitted convictions in notification messages for two specific situations:

- When a notification message provides an update of a conviction previously notified, the notification message needs to refer to the conviction that is actually being modified.

- In the case of the formation of an overall sanction, a notification on this new conviction is sent indicating the formation of the overall sanction. This notification message indicates that this conviction affects other convictions previously notified by referring to them.

Considering the fact that an update of a conviction may occur for convictions that have been notified to the Member State of nationality before NJR or before ECRIS, the notification message must be able to relate to a previously notified conviction by using one of the following:

- For convictions that have already been notified using ECRIS: the unique ECRIS identifier of the conviction, or
- For convictions that have been notified using NJR: the unique NJR identifier of the conviction (called "decision id"), or
- For convictions that have not been notified before or that have been notified by other means of communication: the code and name of the convicting authority, the file number of the conviction and the final date of the conviction. The code of the convicting authority is defined as optional in this structure due to the fact that not all Member States assign such codes.

In order to fulfil these requirements, the properties "MessageDataUpdatedConviction" and "MessageDataOtherAffectedConviction" have been defined as part of notification messages (i.e. in the "NotificationMessageDataType"). These allow establishing a relation with previous convictions that have either been transmitted via ECRIS, via NJR or via other means of communication.

The property "MessageDataUpdatedConviction" is to be used for indicating that the notification provides an update of the conviction being referred to.

The property "MessageDataOtherAffectedConviction" is to be used for indicating that the conviction contained in the notification message affects other convictions such as in the case of the formation of overall sanction.

Please note that the rules and structures defined in these technical specifications imply that a unique ECRIS technical identifier will need to be assigned to each conviction the first time that it is transmitted through ECRIS in a notification message. This applies also to convictions that may have previously been transmitted using other types of technical identifiers through NJR or any other means of communication.

5.3 XSI:TYPE

In the XSD type definitions of the detailed technical specifications, the "xsi:type" attribute is not explicitly defined. It must be noted however that software implementations generating the XML documents from in-memory objects however explicitly generate the "xsi:type" attribute in complex elements so as to know how to serialise/de-serialise properly from in-memory object to XML and vice versa (also called marshalling/un-marshalling).

It must be noted here also that if sample XML is generated manually from the XSD definitions using tools like SoapUI, the "xsi:type" attribute is not included which may cause problems with some implementations which will not be able to detect into which type of object the XML should be de-serialised.

As a more general rule, the usage of the "xsi:null" attribute should be avoided in the ECRIS implementations so as to avoid interoperability issues.

5.4 Technical Validation Rules

Before describing the technical validation rules, please note that implementers **must** first check whether their specific platform properly implements the "SOAP Version 1.2 Part 2: Adjuncts" and in particular chapter §7.5.1 "Behaviour of Requesting SOAP Node". This is important due to the fact that different platforms interpret in a different way the specification.

Taking this under consideration, it is **expected** that an implementation is capable of returning an HTTP status code 500 whenever a fault is returned to the sender.

The following conditions are to be implemented by the ECRIS software and **must** result in a **synchronous technical error** when encountered:

- The message payload is not structurally valid (i.e.it does not validate against the *XSD*)
- The version of a web service operation used by the sender is no longer valid (please refer to the detailed explanation in the "Technical Architecture" document). This must result in a SOAP fault with the code "WrongVersionUsed".
- The operation invoked is not supported. This situation can **only** occur when a sender is invoking a "Push" operation on a receiver that does not support such an operation mode. This must result in a SOAP fault with the code "OperationNotSupported".

Please note that in addition to the error conditions mentioned above which are specific to ECRIS, various other situations resulting in an HTTP 500 status code and a SOAP fault as return message can occur. As a consequence, any given application server or *web service* framework can potentially return such an HTTP 500 status code for various other reasons such as connectivity problems, hardware failure, etc.. It is not in the scope of this document to provide an exhaustive list of all such possible causes.

Additionally, the HTTP client software **must** always perform the required security checks defined in the "Security Analysis" document for validating the HTTPS certificate provided from the requested server. In case any of those checks fails, the client software **must** immediately drop the connection and issue appropriate warnings to the operators of the ECRIS software so that manual action can be taken.

In any case, technical errors need to be logged thoroughly. It is also recommended that implementers communicate their findings to the central entity so as to constitute a technical knowledge base that can be shared with all Member States experts.

5.5 Functional Validation Rules

The following sections define the conditions that **can** lead to **returning a functional error** to the sender of the message when encountered.

First of all, two categories of functional validation conditions need to be distinguished:

- "automatic" functional validation – the conditions can easily be verified in an automated manner by the ECRIS software which then can automatically send the appropriate error code to the sender of the message.
- "manual" functional validation – the conditions can only be verified by human operators; for these cases, the detailed technical specifications only define appropriate error codes that can be used by the Member States' central authorities in case where this error situation is identified when processing the message during the transaction.

It is left at the discretion of each implementer to decide if and in what manner the validation processes are implemented. However each ECRIS software implementation must be capable of at least receiving and understanding the functional error codes defined.

In order to minimise the occurrence of functional errors, it is also recommended that the ECRIS software of the sending Member State applies the functional validation rules described in the next sections before transmitting the message.

For each error condition described in the next sections, the common reference table "FunctionalErrors" defines a corresponding error code to be used in the functional error messages to be returned to the sender of a problematic ECRIS message.

5.5.1 « Automatic » Validation

The evaluation of the following conditions can be automated in the ECRIS software and should lead to a functional error when encountered in an ECRIS message.

General

[GEN-1] The size of the message, including attachments, exceeds 10 megabytes.

(Please note that this size limit for all ECRIS messages has been defined and agreed upon in the "Technical Architecture" document so as to limit the overall volume and network bandwidth to be consumed by the ECRIS data exchanges.)

[GEN-2] One of the following messages does not correlate to a previously received message, meaning that the technical identifier indicated in the property "MessageMetaDataResponseTo" does not match the technical identifier of any previously received message:

- CancellationMessage
- FunctionalErrorMessage
- RequestDeadlineMessage
- RequestResponseMessage
- RequestResponsePushMessage
- NotificationResponseMessage
- NotificationResponsePushMessage
- RequestAdditionalInfoMessage
- RequestAdditionalInfoResponseMessage
- RequestAdditionalInfoResponsePushMessage

[GEN-3] One of the following messages does correlate to a previously received message within the same transaction but is not expected to occur at this moment of the transaction, according to the defined kinematics:

- CancellationMessage
- RequestDeadlineMessage
- RequestResponseMessage
- RequestResponsePushMessage
- RequestAdditionalInfoMessage
- RequestAdditionalInfoResponseMessage
- RequestAdditionalInfoResponsePushMessage
- NotificationResponseMessage
- NotificationResponsePushMessage

Binary Attachments

[BIN-1] The binary file attached is not a NIST file.

[BIN-2] The binary file attached is corrupt or has a different MD5 hash code than the one provided.

"Person" entity

[PER-1] The "surname" property of the person contains a dummy value (i.e. element `dm:PersonType:PersonName:Surname`).

[PER-2] The birth date of the person is equal or after the current date.

[PER-3] The birth date of the alias of the person is equal or after the current date.

[PER-4] The birth date of the person is after the issuing date of the person's identification document.

[PER-5] The birth date of the alias of the person is after the issuing date of the identification document of the alias of the person.

[PER-6] The alias fore- and surname contain identical information as the person fore- and surname.

[PER-7] The birth date is equal or after any of the following dates:

- Decision: decision date
- Decision: final decision date
- Offence: start date
- Offence: end date
- Sanction: execution start date
- Sanction: execution end date
- Sanction: sentenced start date
- Sanction: sentenced end date

"Place" entity

[PLA-1] The town code references a value in the common reference table "Cities" that is in a country that is different than the "country" property of the entity.

[PLA-2] The country subdivision references a value in the common reference table "Country Subdivisions" that is in a country that is different than the "country" property of the entity.

Please note that if the "town name" property does not match the value referenced by the "town code" property, it is assumed that the "town name" has precedence. No functional error is to be returned in this case.

Please note also that if the town name provided is misspelled or spelled in another language than the one expected (e.g. indicating "Saragosa" instead of "Zaragoza", "Parigi" instead of Paris, etc.), it should not lead to an automatic rejection of the message carrying the information. Here also, no functional error is to be returned in this case.

"Identification Document" entity

[IDD-1] The issuing date is after the "valid until" date.

[IDD-2] The issuing date is after the current date.

"Decision" entity

[DEC-1] The decision date is after the final decision date.

[DEC-2] The decision date is after the current date.

[DEC-3] The final decision date is after the current date.

[DEC-4] The decision date is before the decision date of the initial conviction.

"Conviction" entity

[CON-1] The field "non-criminal ruling" is set to "true" and one of the sanctions in this conviction is in one of the following common category (including sub-categories):

- "1000 – Deprivation of freedom"
- "4000 – Prohibition or expulsion from territory"
- "10000 – Military penalty"

[CON-2] The field "non-criminal ruling" is set to "true" and one of the offences in this conviction is in one of the following common category (including sub-categories):

- "0100 00 Crimes within the jurisdiction of the International Criminal Court",
- "0200 00 Participation in a criminal organisation"
- "0300 00 Terrorism"
- "0400 00 Trafficking in human beings"
- "0500 00 Illicit trafficking and other offences related to weapons, firearms, their parts and components, ammunition and explosives"
- "1400 00 Tax and customs offences"
- "1900 00 Forgery of means of payment

[CON-3] The conviction's decision date is after the conviction's final decision date.

[CON-4] The conviction's decision date is after the current date.

[CON-5] The conviction's final decision date is after the current date.

"Offence" entity

[OFF-1] The end date is before the start date.

[OFF-2] The start date is after the decision date of the conviction.

[OFF-3] The start date is after the final decision date of the conviction.

[OFF-4] The end date is after the final decision date of the conviction.

[OFF-5] The start date is equal or after the current date.

[OFF-6] The end date is equal or after the current date.

[OFF-7] The common category references a valid value in the common reference table while the national title indicates the dummy value "UNKNOWN".

[OFF-8] The common category and national title reference dummy values while at least one of the following fields contains a non-dummy value:

- Number of occurrences
- Level of completion
- Level of participation
- Applicable legal provisions
- Place
- Recidivism

"Sanction" entity

[SAN-1] The common category references a valid value in the common reference table while the national title indicates the dummy value "UNKNOWN".

[SAN-2] The number of fines indicates a value > 0 while the sanction category does not reference the category "8000 - Financial Penalty category code" or one of its child categories.

[SAN-3] The sanction category indicates "8000 - Financial Penalty category code" or one of its child categories but the amount of individual fine is not provided (i.e. the XML element indicating the amount of the individual fine is either not present or its value is set to 0).

[SAN-4] The number of fines indicates a value > 0 while the amount of individual fine is not provided.

[SAN-5] The amount of individual fine indicates a value > 0 while the sanction category does not reference the category "8000 - Financial Penalty category code" or one of its child categories.

[SAN-6] The currency of fine references a valid value while both the number of fines and the amount of individual fine are not provided.

[SAN-7] The currency of fine references a valid value while the sanction category does not reference the category "8000 - Financial Penalty category code" or one of its child categories.

[SAN-8] The number of fines and the amount of individual fine contain valid values but the currency of fine is not provided.

[SAN-9] The sanction category indicates "1001 – Imprisonment" but the sentenced duration is not provided (i.e. the XML element indicating the sentenced duration is either not present or its value is set to 0).

[SAN-10] The execution start date is before the decision date of the conviction.

[SAN-11] The execution start date is before the final decision date of the conviction.

[SAN-12] The execution end date is before the decision date of the conviction.

[SAN-13] The execution end date is before the final decision date of the conviction.

[SAN-14] The execution start date is equal or before the offence start date.

[SAN-15] The execution start date is before the offence end date.

[SAN-16] The execution end date is equal or before the offence start date.

[SAN-17] The execution end date is equal or before the offence end date.

[SAN-18] The execution start date is after the execution end date.

[SAN-19] The execution end date is earlier than (execution start date + duration).

[SAN-20] The sentenced start date is before the decision date of the conviction.

[SAN-21] The sentenced start date is before the final decision date of the conviction.

[SAN-22] The sentenced end date is before the decision date of the conviction.

[SAN-23] The sentenced end date is before the final decision date of the conviction.

[SAN-24] The sentenced start date is equal or before the offence start date.

[SAN-25] The sentenced start date is before the offence end date.

[SAN-26] The sentenced end date is equal or before the offence start date.

[SAN-27] The sentenced end date is equal or before the offence end date.

[SAN-28] The sentenced start date is after the sentenced end date.

[SAN-29] The sentenced end date is earlier than (sentenced start date + duration).

[SAN-30] The common category and the national title reference dummy values while at least one of the following fields contains a non-dummy value:

- Alternative
- Specific to minor
- Execution start date
- Execution end date
- Execution duration
- Number of fines
- Amount of individual fine
- Currency of fine
- Suspension
- Interruption

"Suspension" entity

[SUS-1] The end date is before the start date.

[SUS-2] The start date is before the decision date of the conviction.

[SUS-3] The start date is before the final decision date of the conviction.

[SUS-4] The start date is before the start date of the offence.

[SUS-5] The start date is before the end date of the offence.

[SUS-6] The end date is before the decision date of the conviction.

[SUS-7] The end date is before the final decision date of the conviction.

[SUS-8] The end date is before the start date of the offence.

[SUS-9] The end date is before the end date of the offence.

[SUS-10] The end date is before the execution start date of the sanction.

[SUS-11] The suspension type is "b – Partially suspended penalty/measure" or "d – Partially suspended penalty/measure with probation/supervision" while the suspension duration is equal or superior to the duration of execution of the sanction.

[SUS-12] The suspension type is "a – Suspended penalty/measure" or "c - Suspended Penalty/measure with probation/supervision" while the suspension duration is inferior to the duration of execution of the sanction.

Example:
Sanction "1001 –Deprivation of freedom" is 14 months with suspension where the suspension duration is for 8 months while the type of the suspension is not indicated as "partial".

[SUS-13] The end date is earlier than (start date + duration)

[SUS-14] The suspension probation duration has a value > 0 while suspension type indicates "a – Suspended penalty/measure" or "b – Partially suspended penalty/measure".

[SUS-15] The suspended amount of the fine indicates a value > 0 while the number of fines indicated in the related sanction is superior to 1.

[SUS-16] The suspended amount of the fine indicates a value > 0 while the category of the related sanction does not reference the category "8000 - Financial Penalty category code" or one of its child categories.

[SUS-17] The suspended amount of the fine is equal or exceeds the amount of individual fine indicated in the related sanction.

[SUS-18] The suspended amount of the fine indicates a value > 0 while the suspension type is not "b – Partially suspended penalty/measure" or "d – Partially suspended penalty/measure with probation/supervision" (i.e. the suspension is not declared as partial).

“Interruption” entity

[INT-1] The end date is before the start date.

[INT-2] The start date is before the decision date of the conviction.

[INT-3] The start date is before the final decision date of the conviction.

[INT-4] The start date is before the execution start date of the sanction.

[INT-5] The end date is before the decision date of the conviction.

[INT-6] The end date is before the final decision date of the conviction.

[INT-7] The end date is before the execution start date of the sanction.

[INT-8] The end date is earlier than (interruption start date + duration).

Requests

[REQ-1] Request purpose category = “001 – criminal proceedings” while the requesting authority type = “A – competent administrative authority”, “P – person concerned for information on own criminal records” or “E – Employer”

[REQ-2] The new deadline provided in the “request deadline” message does **not** fulfil the following conditions:

- For proceedings requiring a maximum of 10 working days (the requesting authority type is “J – judicial authority” or “A – competent administrative authority”):
  - deadline < (reception of request message + 16 calendar days)
  - deadline < (reception of additional ID info message + 16 calendar days)

  (Contingency used for future date is 1 week-end + 4 office closing days)
- For proceedings requiring a maximum of 20 working days (the requesting authority type is “P – person concerned for information on own criminal records”):
  - deadline < (reception of request message + 34 calendar days)

  (Contingency used for future date is 3 week-ends + 8 office closing days)

Common Reference Tables

[CRT-1] One of the following information elements transmitted does not contain a value from the corresponding common reference table:

- Place: Country → common reference table “Countries and Nationalities”
- Place: Town Code → common reference table “Cities”
- Place: Country Subdivision → common reference table “Country Subdivisions”
- Person: Nationality → common reference table “Countries and Nationalities”
- Sanction: Currency of Fine → common reference table “Currencies”
- Request message: Purpose Category → common reference table “Request Purposes”
- Requesting Authority: Type → common reference table “Requesting Authorities”
- Identification Document: Identification Document Category → common reference table “Identification Document Categories”
- Conviction: Convicting Country → common reference table “Countries and Nationalities”
- Offence: Common Category → common reference table “Offences”
- Offence: Level of Completion → common reference table “Levels of Completion”
- Offence: Level of Participation → common reference table “Levels of Participation”
- Sanction: Common Category → common reference table “Sanctions”
- Sanction: Type → common reference table “Types of Sanction”
- Sanction: Alternative → common reference table “Types of Alternative Sanctions”
- Suspension: Type → common reference table “Types of Suspension”
- Decision Change: Type → common reference table “Types of Decision Changes”
- Request Response message: Other Member State → common reference table “Central Authorities”

[CRT-2] Usage of expired reference values:

- For the “Currencies” common reference table: the “Valid To” information field of the currency value used in currency of fine is before the decision date of the conviction.

Example: A person of Belgian nationality, convicted on 31.08.2014 in Slovakia, is the subject of a notification transmitted by Slovakia to Belgium on 4.10.2014. One of the sanctions transmitted is the payment of a fine of 5000 Slovak koruna. As it can be easily observed, from the Common reference table spread-sheet, this is a functional error as Slovak koruna was valid only up until 01.01.2009.

- For the "Countries" common reference table: the "Valid To" information field of the country value used in the birth place is before the birth date.

  Example: A person, subject of a notification in 2014, is born on 10.02.1993 and has as country of birth the value "Yugoslavia". This represents a functional error, since the value Yugoslavia was valid until 01.01.**1992**.

  Please note that for countries, an entry may be at the same time valid as country of birth of the person referred in the message but not as current nationality. As an example, the value "Czechoslovakia" can be a valid country of birth in a message if the person was born before 1993 but cannot be used in the same message as being the current nationality.

  The same rule is not to be applied to the "Cities" and "Country Subdivisions" common reference tables. Indeed, for these fields such a rule may be counterproductive. If for example a message uses the current name of a city as town of birth whereas the city was named differently on the date of birth of the person, it can still prove useful to keep this information so as to find the person in the national registers.

Logging, Monitoring and Statistics

[LMS-1] An ECRIS message is received for which the timestamp indicates that it was sent during the interval after time 21:59:59 of the last calendar day of the month and before time 02:00:01 of the first calendar day of the month.

According to the "Logging, Monitoring and Statistics Analysis" document, this specific condition must raise a specific functional error code.

5.5.2 « Manual » Validation

The evaluation of the following conditions cannot be automated in the ECRIS software. However when encountering such a situation where the ECRIS message obviously contains an inconsistency, the end-user should be able to reject the faulty message and to respond to the sender with the most appropriate functional error code.

[MAN-1] The information received in a message successfully passes the automated technical and functional validations but the data contained in the information elements is unusable.

A few examples:

- Offence National Title = "GRTSDSRDAH" and Offence Applicable Legal Provisions = "jckzekfhfkfehkfe" and Offence Start Date = "dummy"
- Decision Date = "dummy", Final Decision = "dummy", Convicting Authority Name = "QWERTYU", Convicting Authority Code = "00000", File Number = "abcde/abc"
- A request message containing only dummy values for the mandatory elements
- Person forename = "Driving" and person surname = "Licence"

[MAN-2] The national title of the offence does not match the common category that has been provided.

Example:
Common category = "0301 00 – Directing a terrorist group" and national title = "Polygamy"

[MAN-3] The national title of the sanction does not match the common category that has been provided.

Example:
Common Category = "1001 – Imprisonment" and National Title = "Fine"

[MAN-4] The additional identification information provided does not correspond to the initial identification information provided in the notification or request message.

[MAN-5] A notification message providing an update of a conviction previously notified correlates with a previous notification message but the new conviction information contained does not match at all with the information of the conviction that is supposed to be updated.

Please note that **no functional error** is foreseen for the case where a notification message conveying an update to a conviction does **not** correlate to a previously notified conviction (i.e. meaning that the technical identifier indicated in the property "MessageDataUpdatedConviction" does not match the technical identifier of any previously notified conviction – identified either through the unique ECRIS identifier of the conviction, the unique NJR identifier of the conviction or a structure formed of the code and name of the convicting authority, the file number of the conviction and the final date of the conviction). Indeed, following the snapshot approach already defined in the "Business Analysis" document, a notification message is always complete and contains all up-to-date information about the conviction. In this specific case where the notification message indicates that it provides an update but that the initial conviction is not known by the receiver, the notified Member State should simply treat the message as if it was the notification of a new conviction.

5.6 Guidelines and Recommendations

5.6.1 Implementation Platforms

As mentioned earlier, the technical artefacts of the detailed technical specifications have been tested using various development platforms. These platforms have been chosen based on their maturity, market penetration and preferences indicated by the Member States in the "Inception Phase Questionnaire".

It is necessary to underline that the technical artefacts that constitute the ECRIS detailed technical specifications (i.e. the WSDL and XSD files) are **fully compliant** with the specifications established in the "Technical Architecture" document and are **purposefully not designed for accommodating specific development platforms**. The aim of the technical artefacts is to stick at 100% to the specifications defined in the "Technical Architecture" document so as to remain neutral towards the development platforms.

Tested platforms

Concretely, the following table provides an overview of the platforms used with their according versions:

Platform	Platform / Vendor URL	Version
Apache AXIS1	http://ws.apache.org/axis/	1.4
Apache AXIS2	http://ws.apache.org/axis2/	1.5.2
Apache CXF	http://cxf.apache.org/	2.3.0
Spring WS	http://static.springsource.org/spring-ws/sites/1.5/	1.5.9
JAX-WS RI (Metro Core)	https://jax-ws.dev.java.net/	2.2.1
.NET WCF	http://msdn.microsoft.com/en-us/netframework/aa663324.aspx	4.0

Table 4 – Implementation platforms and their versions

The operating system used for the tests is Microsoft Windows version 6.1.7600.

The *JDK* that has been used so as to program and compile the Java classes produced by the Java-related platforms is version 1.6.0_13. In addition, the Apache Maven software project management and comprehension tool version 3.0 has been used to handle automated builds and to support basic testing.

For the development and testing of the .NET WCF platform, Visual Studio 2010 version 10.0.30319 has been used. The artefacts for .NET were generated using "svcutil" version

3.0.4506. In addition, the plug-in WCF Blue (http://wscfblue.codeplex.com/) for Visual Studio was also used to generate the artefacts within Visual Studio.

Main conclusions

Overall, the tests performed allowed to proactively fix possible incompatibilities of the tested platforms with the technical artefacts. It must be noted here that the term "incompatibilities" does not refer to the technical artefacts but solely to the development platforms. Indeed, each development platform is a software product implementing the *web services* specifications. Thus, the quality of the implementation in regards to the specifications provided by *W3C* for *web services* depends solely on how the developers that were involved in that task perceive the said implementations. Please note also that these incompatibilities spotted during practical testing are not blocking for properly producing and consuming the ECRIS messages but may require specific minor adaptations to the development platform itself in order to by-pass these minor issues.

Beyond testing for incompatibilities, during this process the platforms were also evaluated for their extensibility and compatibility. Based on that evaluation, the following recommendations can be made:

- It is strongly recommended that implementers update their implementation platforms up to the latest available and stable version.

  Using the latest stable version of a given development platform will ensure that common issues and well-known bugs have been fixed and thus this platform will implement more closely the technical specifications to be used.

- For the JAVA related frameworks, it is recommended that implementers consider Spring WS and Apache CXF for their implementations.

  Indeed AXIS1 provided its latest delivery on 22 April 2006 and can safely be considered a deprecated project. AXIS2 is a mature and stable framework but the releases provided up to now are mainly maintenance releases with very long release cycles. Version 1.5.1 was released on 23 October 2009 whereas the latest stable version 1.5.2 was released only on 06 September 2010. The JAX-WS RI provided by Oracle provides only basic functionality and the integration and extensibility options that are supported are minimal.

  Apache CXF and Spring WS on the other hand are actively maintained, integrate easily with other technologies and are easy to extend without requiring substantial effort. Additionally both frameworks are open source, with support from large communities and the available knowledge base on Internet for is quite significant.

  Regardless of the recommendations made above, please note that all JAVA platforms that implement properly the specifications, standards and protocols defined in the "Technical Architecture" document will not have problems in consuming or producing messages based on the technical artefacts of ECRIS.

- If automatic tools are used so as to generate code from the technical artefacts, it is also recommended that implementers perform thorough reviews of the produced code so as to ensure their validity.

  This recommendation is based on the fact that during the testing, some tools have been found to interpret and generate the specifications in an erroneous way. This is the case mostly for the Axis platforms, especially Axis1 which faced difficulties in producing automatically code that would compile. Axis2 on the other hand, on version 1.5.1 when using "adb" for serialisation, could not produce classes for type definitions that were deriving by restriction from their parent types. This issue could be avoided by moving to a newer version of Axis2.

In addition, attention should be paid to the service interfaces that are generated by the automated tools, especially due to the fact that the operations defined in the *service contract* are "one-way" operations but can potentially return faults. It has been noted that some tools like Apache WCF or JAX-WS RI flag such operations using annotations or attributes which explicitly define them as being "one-way". In doing so, the frameworks tend to "swallow" faults returned by the application code rather than returning them to the caller.

Impacts on *service contract*

Lastly, it must be noted that specific decisions on some details of the design of the *service contract* have been taken so as to mitigate problems that some development platforms encounter when processing it.

More concretely, the optional "name" attribute of the element "<wsdl:input>" must be used and must contain identical values, both under the "portType" and "binding" definitions of the document. This was necessary so as to avoid a well-known problem that "svcutil" from the .NET framework encounters when processing *WSDL* files which define a different name in the "portType" "<wsdl:input>" element than the one used in the "binding" (for more information on that subject, please refer to http://social.msdn.microsoft.com/Forums/en-US/wcf/thread/626bde79-45c4-4ae2-ba7d-dd7834bacd60).

Also, even though the operation "isAlive" should follow the "notification" primitive defined in the *WSDL* specification 1.1 (see also http://www.w3.org/TR/wsdl#_notification) it was not possible to define the service using the appropriate grammar because Apache CXF would not support it. For that reason, the operation has been defined as requiring an input message. This input message can however be ignored by the ECRIS software implementations.

5.6.2 "isAlive" Operation

As defined in the "Technical Architecture" and repeated in this document, the "isAlive" operation allows a Member State to check whether another Member State's end-point can be reached and whether it is properly responding. Implementers of this operation can essentially use it as a flag signaling whether their ECRIS system can accept and process appropriately messages.

Even though no strict policies are applied on the *service contract* or defined in this document in regards to the invocation of this operation, implementers should take under consideration the following suggestions:

- The "isAlive" service should be invoked within regular intervals so as to monitor the availability of other ECRIS implementations. The regular intervals should range between 6 and 12 hours so as to allow taking corrective measures in case an ECRIS instance is found to be unavailable.
- It is likely that several Member States will first pile up a series of notifications before sending them all out during a predetermined moment later in time. This results thus in a significant amount of messages being sent during a specific and predetermined period of time. It is thus recommended to the implementers to use the "isAlive" operation **once before** initiating such a **grouped sending of messages**. Indeed, in the case that the receiving software is unavailable, following this recommendation will greatly reduce the number of failed messages exchanged as well as reduce the resource usage of the sending Member State. Similarly, the "isAlive" operation should not be systematically invoked before sending each business message so as to avoid multiplying unnecessarily the number of messages.
- The implementation of the "isAlive" service should not only indicate whether the web service end-point alone is up and running. It should also check the state of the back-end systems participating in the processing of the ECRIS messages and should return a negative message if the chain of systems cannot completely process the messages.

A positive side effect of using the "isAlive" service in the ways suggested above is the fact that other errors, not directly related with the availability of the requested ECRIS instance, can be spotted and can allow taking corrective measures proactively. For example, the "isAlive" service must reply with a "soap:fault" using the "WrongVersionUsed" code in case the calling ECRIS end-point is using an older version of the service than the one it should use. If such a fault is detected, the calling ECRIS instance should immediately log the incident and the system operators should be notified so as to rectify the problem.

5.6.3 Retry Policies

It is a common practice in message exchanges to retry a message transmission if it has failed in a previous call, especially when the cause of the failure related to the network connectivity. Such retries are not explicitly defined by the detailed technical specifications, since such policies are specific to the implementations and are not declared in the *service contract*. In particular, it is **not** required for an ECRIS instance to necessarily support such policies.

The reasons a message exchange could fail in ECRIS have to be viewed both from a technical and from a business perspective. The difference between the two perspectives is that the business retransmissions are usually foreseen as part of the process, if for example a business error has occurred. On the other hand, the technical perspective usually foresees automatic retransmission of messages in case of technical failures. These cases include general availability problems (i.e. connectivity, ECRIS instance replied with "false" during an "isAlive" operation, etc.) as well as technical errors provoked by the receiver, such as considering a message structurally invalid whereas the sender has already structurally validated the message or the HTTPS certificate provided by the server was found invalid.

On the technical level, only **availability issues** should lead to **automated retransmission** of failed messages. Indeed, technical problems provoked from the receiver, such as considering a structurally valid message as invalid, usually requires human intervention in order to be rectified.

In addition, it is also advisable that implementers follow common practices for the automated retransmission policies. It is strongly suggested **not to immediately retry** sending a failed message, but rather to keep a queue or back-log of failed messages and retry at a later moment in time. It is also advised to increase the time interval with each attempt. Concretely, if a first attempt to resend a failed message occurs 1 hour after the initial transmission, it is recommended that the next attempt is made only after 3 hours. Following this practice should again reduce the number of failed messages. Also, the time interval between retries should be sufficiently large so as to allow system operators to rectify the problems that blocked the initial message transmission.

5.6.4 Traffic Management

Please note that at the time of writing of this document, no reliable figures are available on the expected network traffic or number of messages to be exchanged after the go-live of the ECRIS implementations in April 2012.

Obviously, the number of messages exchanged varies greatly from Member State to Member State due to specific factors such as the size of the country, the number of borders with other Member States, the movements of people of various nationalities, etc. However, from the answers received to the "Inception Phase Questionnaire" and from the intermediate NJR statistics of 2010, it can be observed that the **current** maximum number of notifications and requests exchanged ranges up to approximately 96.000 per year for one Member State. Considering the fact that in ECRIS, at least 2 *web service* operations are required for completing properly a notification transaction and at least 3 are required for completing a request transaction, the current volume of exchanges would already imply that that Member State would need to process at least approx. 16.000 *web service* operations per month.

The figures above are however only representing a very rough extrapolation of the current situation. The fact that a significant increase of message exchanges is expected by most Member States once ECRIS is operational also needs to be taken into account. This expected increase is mainly due to the facts that all EU Member States will be participating in the exchanges and that ECRIS provides support for handling requests for non-criminal proceedings. In addition to this, the currently on-going project ECRIS-TCN (Third Country Nationals) is also exploring the possibility to leverage in the future the available functionality of ECRIS for realising additional criminal records information exchanges on non-EU citizens.

Based on the figures above, and taking a large buffer margin, it can be reasonably assumed that the number of *web service* operations could range up to 50.000 per month for a given Member State.

Please note that during the tests performed when designing and implementing the detailed technical specifications, the approximate size observed for an XML message without binary attachment is of 6Kb. As a reminder, the maximum size has been limited to 10 Mb in the "Technical Architecture" document.

Given the figures above, solid and mature frameworks and systems should not be easily overwhelmed by the expected amount and volume of messages. However, difficulties may occur when high peaks of messages are sent and/or received.

Thus, in order to mitigate these risks, the following recommendations can be made:

- On the receiver's end, it is recommended to establish priorities for the processing of incoming messages. Generally, messages related to the "request" process should be treated with higher priority than messages related to "notifications". Within the group of "request"-related messages, it is also recommended to put the highest priority on the processing of the "Request" and "Request Deadline" messages.
- On the sender's end, it is also strongly recommended that ECRIS implementations do **not** send all the volume of their messages at the same time of the day but rather prioritise the messages, following the same priorities as defined above, and send them in smaller groups at regular intervals.
- In order to establish more accurate figures in the future for performing traffic management, implementers are advised to monitor and establish statistics on the network bandwidth used, sizes of the files being exchanges and number of messages per type.

6 FUTURE EVOLUTIONS

This chapter briefly provides an overview of the approach that to be followed when creating a new version of the technical artefacts.

Please note that a detailed description on the versioning strategy and its implementation in the context of ECRIS has already been provided in the "Technical Architecture" document. This chapter does not aim to alter in any way the approach already agreed upon but rather to make more explicit how it should be performed in practical terms.

6.1 Versioning of WSDL/XSD Files

Each of the technical artefacts already implements the versioning concepts. All files have been suffixed with the version identifier "v1.0". The namespaces defined in the *XSD* and the *WSDL* files are also suffixed with the same version identifier. The definitions of the operations and messages provided in the *WSDL* are also suffixed with the same version identifier.

Assuming that a minor version of these technical specifications needs to be produced, the following actions should be performed by the central entity:

- Implement the changes in the *XSD* and *WSDL* files and suffix these files with the incremented version identifier.
- Suffix all the namespace declarations in the *XSD* and *WSDL* files using the new version number.
- Create a copy of the messages, bindings, operations and service declarations within the *WSDL* document and suffix them accordingly using the new version identifier (the previous declarations with the old version must remain in order to support two concurrent versions).
- If necessary, remove the oldest definitions of messages, bindings, operations and service declarations from the *WSDL* document (only if the WSDL would otherwise support more than two concurrent versions).
- Document and validate the changes performed in the technical artefacts.
- Provide the new versions of the technical artefacts to the Member States' experts for revision.

Once the new version of the technical artefacts is agreed upon, and once a Member State has implemented this new version in its national ECRIS software, the Member State must modify the newest version of the *WSDL* file and update the "location" attribute of the "soap12:address" for each version of the "wsdl:service" element so as to properly reflect the *HTTP* address that other Member States should use in order to send ECRIS messages. These updated service contracts must be sent to the central entity, along with any other information that can affect the other Member State (such as the first date when the new implementation will be deployed or whether there was a change in the implementation of binary attachments).

As a final step, the central entity validates the newly received *WSDL* file and disseminates it to all other Member States. During this validation, the central entity should verify at least the following points:

- that the WSDL file version and version information used in the namespaces are still correct
- that the WSDL file has not been altered in the definition of messages and operations
- that the end-point declared in the WSDL file indeed corresponds to an end-point that is accessible through the sTESTA network

6.2 Changing Common Reference Tables

Please note that the rules to be strictly applied when changing the content of common reference tables have already been detailed in the "Technical Architecture" document.

In addition, the technical approach to be followed when modifying the common reference tables is similar to the one described earlier for producing new versions of the other technical artefacts. Changing either **the structure or the content** of the common reference tables **must always** result in a new minor version. This approach is preferred because it does not require additional kinematics for establishing which version of the common reference tables is supported by a specific ECRIS software implementation.

7 ANNEX – OVERVIEW OF MESSAGE VOLUMES

7.1 Overview of Member States' Answers

- The following provides an overview on the answers provided by the Member States' central authorities to the specific question of the "Inception Phase Questionnaire" on the expected volume of information that would need to be exchanged, assuming that ECRIS would be used currently:
  - Approx. 96.000 notifications and requests per year: PL
  - Approx. 55.000 notifications and requests per year: FR
  - Between 25.000 and 50.000 notifications and requests per year: ES, UK
  - Between 10.000 and 25.000 notifications and requests per year: BE, CZ, LT, LU, SE
  - < 10.000 notifications and requests per year: EE, FI, HU, RO, SI, SK
  - Unknown/not provided: AT, DE, NL, PT
- Do you foresee an increase of exchanges of criminal record data with other Member States once the ECRIS software is operational in all Member States?
  - Yes: AT, BE, CZ, ES, FI, FR, HU, LT, LU, NL, PL, PT, RO, SE, SK, UK
  - No: DE, SI
  - Unknown: EE

7.2 Overview of NJR Statistics 2010

The following figures have been extracted from the consolidated statistics of the NJR message exchanges for the period ranging from January until July 2010, thus 6 months of activity.

Member State	Notifications	Requests	Total
BE	5862	2457	8319
BG	2131	1444	3575
CZ	4708	2943	7651
DE	22089	18699	40788
ES	5652	1736	7388
FR	12134	7998	20132
IT	1944	228	2172
LU	3313	1712	5025
PL	14468	9988	24456
SK	3037	1877	4914

Please note that this table indicates for notifications and requests both the messages **sent and received** by the NJR Member State during the defined period of time. The column providing the total values indicates simply the sum of notification and request messages sent and received by each Member State during the given period of time.

This implies in particular that a given NJR message is counted twice in this table: once for the Member State that has sent it and once for the Member State that has received it. This is correct because this table mainly aims at showing the order of magnitude of the number of messages to be treated for each Member State, rather than providing exact counts of the NJR messages.

____

COUNCIL OF THE EUROPEAN UNION	**Brussels, 19 October 2011**
	11275/1/11 REV 1
	LIMITE
	COPEN 151 EJUSTICE 52 JURINFO 42

NOTE

from:	General Secretariat of the Council
to:	Delegations
Subject:	ECRIS Detailed Technical Specifications

Delegations will find in the Annex the revised text of the ECRIS Detailed Technical Specifications as resulting from the observations submitted by the Member States and examined during the Expert Meeting on 21 September 2011. Changes are indicated by underlined text as against the previous version of the ECRIS Detailed Technical Specifications.

ANNEX

European Commission – DG Justice

iLICONN Consortium (Bilbomatica – Intrasoft – Unisys)

ECRIS Technical Specifications

Detailed Technical Specifications

Document Information

AUTHOR	iLICONN – Intrasoft International S.A.
OWNER	European Commission – DG Justice
ISSUE DATE	04/10/2011
VERSION	1.6
APPROVAL STATUS	Final

Authors

NAME	ACRONYM	ORGANISATION	ROLE
Nicholas YIALELIS	NYI	iLICONN – Intrasoft International S.A.	Manager Reviewer
Panos ATHANASIOU	PAT	iLICONN – Intrasoft International S.A.	Main Author
Ludovic COLACINO DIAS	LCO	iLICONN – Intrasoft International S.A.	Contributor Reviewer

Document History

VERSION	DATE	AUTHOR	DESCRIPTION
0.1	29/10/2010	LCO	First draft
0.2	03/11/2010	PAT	Updates on all sections
0.3	05/11/2010	PAT	Updates on all sections
0.4	07/11/2010	PAT	Minor updates in various sections
0.5	10/11/2010	LCO	Revision
0.6	24/11/2010	PAT	Update for implementing comments
0.7	25/11/2010	LCO	Revision for delivery of author's position
0.8	17/12/2010	PAT	Implementation of comments and changes agreed during the Expert Group Review meeting 01-Dec-2010 and COPEN meeting 10-Dec-2010
0.9	20/12/2010	LCO	Revision before final delivery
1.0	22/12/2010	LCO	Final delivery
1.1	23/02/2011	DCO	Implementation of comments and changes agreed during the Expert Group Review meeting 16-Feb-2011
1.2	24/02/2011	PAT	Implementation of comments and changes agreed during the Expert Group Review meeting 16-Feb-2011
1.3	28/02/2011	PAT	Minor corrections
1.4	28/02/2011	LCO	Final revision
1.5	01/03/2011	DCO	Minor corrections after revision by DG JUST
1.6	04/10/2011	DCO, PAT, LCO	Implementation of changes agreed during the Expert Group Review meeting 21-Sep-2011 Correction of additional discrepancies that have been identified

TABLE OF CONTENTS

1 DOCUMENT

1.1 Purpose

This document is a formal product of the *ECRIS Technical Specifications* project for the European Commission – DG Justice and produced by the iLICONN Consortium.

The main purpose of this document is to describe the technical artefacts that provide the detailed technical specifications of the ECRIS data exchanges. It presents the logic, rules and detailed content of the technical artefacts constituting the *ECRIS Technical Specifications* and aims at providing all necessary information and guidelines for the developments teams that will need to implement these technical specifications.

This document assumes that the readers have a good and detailed knowledge and understanding of the following elements:

- The ECRIS legal basis (Council Framework Decision 2009/315/JHA and Council Decision 2009/316/JHA)
- The "ECRIS Technical Specifications – Inception Report" document
- The "ECRIS Technical Specifications – Technical Architecture" document
- The "ECRIS Technical Specifications – Security Analysis" document
- The "ECRIS Technical Specifications – Business Analysis" document
- Detailed knowledge and understanding on IT-technical level of *Web Services*, *XSD* and *XML*

1.2 Scope

This document provides all necessary background information so as to gain a deep understanding of the technical artefacts that define the ECRIS *Detailed Technical Specifications*. In particular, this document provides:

- the general descriptions of the WSDL, XSD and XML files provided
- the general logic, rules and guidelines that have been followed for designing and defining these technical artefacts
- detailed descriptions of the design choices
- UML class diagrams depicting in a logical form the relations between the XSD entities that are defined in the technical artefacts
- UML sequence diagrams illustrating the sequences in which the *web service* operations are meant to be used
- roles of the involved stakeholders for the management and future evolutions of the technical artefacts
- the minimum set of technical and logical validation rules that must be implemented in the ECRIS software
- recommendations and guidelines for the implementers of the detailed technical specifications

This document does not provide any other information than what has been stated above, and in particular it does not include:

- a copy of the content of the *WSDL*, *XSD* and *XML* files so as to avoid redundancy; it is expected that the reader opens these technical files into appropriate reading/editing software in addition to reading this document
- the overall technical system architecture (please refer to the "Technical Architecture" document)
- the functional and business concepts (please refer to the "Business Analysis" document)
- the IT-security considerations related to the ECRIS data exchanges (please refer to the "Security Analysis" document)

1.3 References

The following sources have been used as input for the elaboration of this "Technical Architecture" document:

[1] ECRIS Legal Basis – Council Framework Decision 2009/315/JHA

Council Framework Decision 2009/315/JHA of 26 February 2009 on the organisation and content of the exchange of information extracted from the criminal record between Member States (OJ L 93/23 of 07.04.2009)

[2] ECRIS Legal Basis – Council Decision 2009/316/JHA

Council Decision 2009/316/JHA of 6 April 2009 on the establishment of the European Criminal Records Information System (ECRIS) in application of Article 11 of Framework Decision 2009/315/JHA (OJ L 93/33 of 07.04.09)

[3] Network of Judicial Registers (NJR) – Technical References – version 1.3a (approved) of 13 March 2008

[4] Network of Judicial Registers (NJR) – Technical References – version 1.4 (draft) of 23 November 2009

[5] Network of Judicial Registers (NJR) – *XML* Listings – version 1.4 (final) of 01 July 2009

[6] NJR *WSDL* and *XML* Files v1.4.2 of 21 January 2009 (final)

"CommonTables_and_XML_rel1-4-2_20090121.zip" file containing:

- RegisterService-1.4.2.wsdl (version 1.4.2)
- common.xsd (version 1.4 of 18 December 2008)
- CommonTables-1.3.xsd (version 1.3)
- CommonTables-1.4.2.xml (version 1.4.2)
- error.xsd (version 1.4 of 02 November 2005)
- information.xsd (version 1.4 of 02 November 2005)
- notification.xsd (version 1.4 of 22 November 2005)
- receipt.xsd (version 1.4 of 02 November 2005)
- request.xsd (version 1.4 of 02 November 2005)

[7] NJR *WSDL* and *XML* Files v1.5 (draft)

- RegisterService-1.5.wsdl (draft version 1.5 of 11 August 2010)
- common.xsd (draft version 1.5 of 10 June 2010)
- CommonTables-1.5.xsd (draft version 1.5)
- CommonTables-1.5.xml (draft version 1.5.0)
- error.xsd (draft version 1.5 of 10 July 2010)
- information.xsd (draft version 1.5 of 10 July 2010)
- notification.xsd (draft version 1.5 of 10 July 2010)
- receipt.xsd (draft version 1.5 of 10 July 2010)
- request.xsd (draft version 1.5 of 10 July 2010)

[8] European Commission – DG Enterprise (2004): IDA Architecture Guidelines for Trans-European Telematics Networks for Administrations, version 7.1 of 13 February 2004 (and annexes)

[9] ECRIS Technical Specifications - Inception Report v1.02 of 22 October 2010

[10] ECRIS Technical Specifications – Glossary v1.3 of 07 November 2010

[11] ECRIS Technical Specifications – Technical Architecture v1.0 of 22 October 2010

[12] ECRIS Technical Specifications – Security Analysis v1.0 of 22 October 2010

[13] ECRIS Technical Specifications – Business Analysis v1.6 of 30 September 2011

[14] ECRIS Technical Specifications – Logging, Monitoring and Statistics Analysis v1.5 of 01 March 2011

[15] ECRIS Technical Specifications – Common Reference Tables v0.20 of 30 September 2011

[16] The replies from the following Member States to the "Inception Phase Questionnaire" v0.05 (listed in alphabetical order):

Austria (AT), Belgium (BE), the Czech Republic (CZ), Estonia (EE), Finland (FI), France (FR), Germany (DE), Greece (GR), Hungary (HU), Italy (IT), Lithuania (LT), Luxembourg (LU), the Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), the United Kingdom (UK)

[17] "SOA Design Patterns"

Thomas Erl - Prentice Hall/PearsonPTR (ISBN: 0136135161)

[18] "Web Service Contract Design and Versioning for SOA"
Thomas Erl, Anish Karmarkar, Priscilla Walmsley, Hugo Haas, Umit Yalcinalp, Canyang Kevin Liu, David Orchard, Andre Tost, James Pasley - Prentice Hall/PearsonPTR (ISBN: 013613517X)

[19] "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions"
Gregor Hohpe, Bobby Wolf - Addison-Wesley Professional
(ISBN-13: 978-0321200686)

[20] "Creating flexible and extensible *XML* schemas"
Ayesha Malik, 01 Oct 2002
http://www.ibm.com/developerworks/library/x-flexschema/

[21] Guide to Versioning *XML* Languages using *XML* Schema 1.1
David Orchard (2006 W3C)
http://www.w3.org/TR/xmlschema-guide2versioning/

[22] ECRIS-TCN Study: Final report of "Feasibility Study: Establishment of a European Index of Convicted Third Country Nationals", 11 June 2010

1.4 About this Document

1.4.1 Elaboration of this Document

This document has been drafted by the iLICONN staff based on the following input:

- The documents listed in the references above
- Information gathered during the preliminary on-site visits of the following Member States' central authorities:
  - 19-Jul-2010 / 30-Jul-2010: Belgium – Service Public Fédéral Justice – Service Casier Judiciaire Central
  - 26-Jul-2010 : France – Ministère de la Justice – Casier Judiciaire National
  - 29-Jul-2010 : Germany – Bundesamt für Justiz – Bundeszentralregister
  - 05-Aug-2010 : United Kingdom – Association of Chief Police Officers (ACPO) – ACPO Criminal Records Office (ACRO)
  - 09-Aug-2010 : Spain – Ministerio de Justicia – Registro central de penados y rebeldes
- The answers provided by the following Member States' central authorities to the questions defined in the "Inception Phase Questionnaire" document that has been sent out by the European Commission to all Member States' contact points on the 04th of August 2010 (listed in alphabetical order):

  Austria (AT), Belgium (BE), the Czech Republic (CZ), Estonia (EE), Finland (FI), France (FR), Germany (DE), Greece (GR), Hungary (HU), Italy (IT), Lithuania (LT), Luxembourg (LU), the Netherlands (NL), Poland (PL), Portugal (PT), Romania (RO), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE), the United Kingdom (UK)

- Direct contacts and meetings with various experts (various experts from the European Commission, experts from the contractor currently developing the *NJR Reference Implementation* software but also other experts that have been involved in various studies and similar projects in the field of justice and cooperation in criminal matters).
- Concrete examples of NJR "notifications", "requests" and "information" messages provided by the following Member States: BE, FR, ES and UK.
- The 263 comments issued by the Member States and the European Commission on the previous version of this document until 19 November 2010.
- The discussions and agreements that have been reached during the Expert Group Review meeting on 01 December 2010 and COPEN Working Party meeting on 10 December 2010.
- The comments issued by the Member States and the European Commission on the previous version of this document.
- Conference calls held with experts from DE, ES, FR and PL between 7 and 9 February 2011 and direct e-mail contacts with several other Member States for discussing the aforementioned comments.
- The discussions and agreements that have been reached during the Expert Group Review meeting on 16 February 2011.
- The discussions and agreements that have been reached during the Expert Group Review meeting on 21 September 2011.

1.4.2 Understanding this Document

This document comes with a "Glossary" document that provides definitions for the specific terms that are used throughout the *ECRIS Technical Specifications* project. By convention, all words marked in italic in this document can be looked up in the "Glossary" document. The bold font is used for emphasising a specific term or part of a sentence. The underlines mark the text that has been added or modified since the last version while the strike-through marks the text that has been removed or replaced. In case of doubts about the exact meaning of a term, please consult first the "Glossary".

In addition, this document comes with the technical artefacts that constitute the *ECRIS Detailed Technical Specifications* to be implemented, in form of *WSDL*, *XSD* and *XML* files:

- ecris-service-v1.0.wsdl
- messages-v1.0.xsd
- domain-model-v1.0.xsd
- common-reference-tables-v1.0.xsd
- common-reference-tables-v1.0.xml
- commons-v1.0.xsd

Should you still have any doubts about the meaning of a specific sentence or paragraph, please do not hesitate to take direct contact with the following persons by telephone or via e-mail, at your best convenience:

Organisation:	European Commission – DG Justice – Criminal Law
Name:	Jaime LOPEZ-LOOSVELT
E-mail:	JUST-CRIMINAL-RECORD@ec.europa.eu
Telephone:	+32 (0)2.298.41.54
Organisation:	iLICONN Consortium – Intrasoft International S.A.
Name:	Ludovic COLACINO DIAS
E-mail:	ECRIS-Specs-PM.iLICONN@intrasoft-intl.com

Mobile: +32 (0)498.30.25.55

1.4.3 Providing Comments

As described in the "Inception Report" document, all major deliverables produced by the iLICONN Consortium are undergoing a "Review Cycle" during which all EU Member States experts are invited to provide comments.

Commenting this document

Since the iLICONN staff needs to collect, compare and analyse the feedback from 27 Member States on the same document – thus potentially a large number of comments – it uses a tool that allows easily extracting the comments from MS Word documents.

Therefore, for commenting **this** document, please apply the following guidelines:

- All comments are to be written in plain English. Comments provided in other languages cannot, unfortunately, be taken into account.
- The comments must be specific to and must relate to the text (sentence and/or paragraph) being revised.
- Please use simple wording and be as specific, concise and clear as possible in order to avoid ambiguities.
- When referring to specific terms, acronyms, abbreviations that are common in your daily jargon but that are not defined in the *Glossary* document, please define them first.
- Write your comments directly in this MS Word document, by proceeding as follows:
  - First select a word, a part of a sentence or a paragraph (this can be done for example by double-clicking on a word or by dragging your mouse over parts of the text while keeping the left mouse-button pressed).

    Attention:

    Please note that a **minimum of 4 characters** must be selected in order for our commenting tool to grab the comment. Furthermore, comments on diagrams and embedded pictures are also not taken into account. In such cases, please select the caption text underneath the diagram or image.
  - Once a word, part of a sentence or paragraph has been selected, insert an MS Word comment in which you can type your remarks.

    An MS Word comment is typically displayed as a red balloon in the right margin of the document and usually starts with the abbreviation of your name and the timestamp at which the comment is being written. Depending on your version of MS Word, use the following steps for inserting a comment:

    MS Word 2007 and MS Word 2010:

    1. Select the text you would like to comment upon
    2. Open the **Review** ribbon, select **New Comment** in the **Comments** section
    3. In the balloon that appears in the right margin, type your comment
    4. Click anywhere in the document to continue editing the document

    MS Word 2003:

    1. Select the text you would like to comment upon
    2. From the **Insert** menu, select **Comment** (or click on the **New Comment** button on the **Reviewing** toolbar)
    3. In the balloon that appears in the right margin, type your comment
    4. Click anywhere in the document to continue editing the document

The text will have coloured lines surrounding it, and a dotted coloured line will connect it to the comment. To delete a comment, simply right click on the balloon and select **Delete Comment**.

- Please do not use the MS Word "track changes" tool and do not write your comments as plain text in the MS Word file.
- In case that you want to provide general comments or remarks that are not specific to a part of the text of this document, please provide them into a separate document and/or e-mail.

In case that you need to translate this document to another language, and then translate back your comments to English, please make sure that your comments are provided in the form described above and that they have not been altered or moved to another section of the text during the translation process.

Commenting the technical artefacts

The *WSDL*, *XSD* and *XML* files that are joined to this document can unfortunately not be commented using the same approach as for the MS Word files. Thus, for providing comments to these files, please use the following guidelines:

- General comments affecting **globally** one or more of the technical files: the general comment can be provided in this document in the appropriate sections where the technical files are listed or described.
- Specific comments that affect **specific** elements within the technical files, such as comments on specific *web service* operations, specific *XSD* elements, specific documentation within the technical artefacts, etc. are to be provided in a separate "Comment Inspection Sheet" spread-sheet. An empty MS Excel file is provided together with this document and with the technical artefacts. This file is to be filled in with the following information for each specific comment:
  - Column "#": please indicate here a unique number for your comment
  - Column "File Name": please indicate here the exact name of the WSDL, XSD or XML file to which the comment applies
  - Column "Rvwr": please indicate here the 2-letter code of your Member State
  - Column "Initial Ver.": please indicate here the version of the WSDL, XSD or XML file to which the comment applies (the version is located in the technical files at the top)
  - Column "Original comment & suggestion by reviewer": please indicate here the comment, by first indicating the part of the file concerned (i.e. name of web service operation, name of XSD element, etc.) and then entering the text of the comment

  It is expected that **one comment** is provided **per line** in the spread-sheet. Please avoid typing several comments in the same line, even if they refer to the same technical file or same technical element within the file. This is asked in order to avoid overlooking or forgetting to process comments accidentally.

2 INTRODUCTION

The "Detailed Technical Specifications" are defining the technical ECRIS *service contract*, which consists of a *WSDL* file defining the *web service* operations, several *XSD* files providing a technical implementation of the domain model that has been defined in the "Business Analysis" as well as an XML representation of the common reference tables.

The principles and approaches regarding the design of these technical artefacts follow the ones defined in the "Technical Architecture" document. The XSD type definitions which determine the domain model are split into different namespaces so as to set-up a meaningful separation of concepts. This separation of namespaces using different files is referred to as "packaging" and each pair of "namespace"-"*XSD*" file is named a "package" throughout this document.

The packaging results in the following *XSD* files:

- **"commons-v1.0.xsd":** contains various common definitions of types which are used from other packages
- **"domain-model-v1.0.xsd":** provides the definitions for the main business entities
- **"common-reference-tables-v1.0.xsd":** contains the definitions of the structures required for the *XML* representation of the common reference tables
- **"messages-v1.0.xsd":** provides the definitions of the ECRIS messages that are sent between Member States' central authorities using the *web service* operations
- **"common-reference-tables-v1.0.xml":** provides the *XML* representation of the common reference tables presented in the "Business Analysis" document, with the addition of technical elements (such as fault codes or types of responses) and functional error codes and descriptions

The WSDL file defining the ECRIS web service operations is the following: **"ecris-service-v1.0.wsdl"**

Ensuring a high level interoperability between the platforms of the implementers is a key element of the ECRIS detailed technical specifications. Therefore the *service contract* for ECRIS incorporates **only** those standards that have been defined in the "Technical Architecture" document.

Please note that, as specified in the "Technical Architecture" document, these technical artefacts are thus solely based on the WSDL 1.1 and XSD 1.0 specifications. In particular, specific standards as *RelaxNG, Schematron* and *XSD 2.0* have been discarded for avoiding interoperability issues or implementation constraints. In the same fashion, *web service* extensions such as *WS-Addressing* are also not supported.

Additionally the technical artefacts have been tested using *WSDL*-to-code generators from some of the most popular platforms to ensure compliance. Also the "*WS-I* Basic Profile 1.1" recommendations have been taken under consideration in order to further ensure interoperability.

3 STAKEHOLDERS, ROLES AND RESPONSIBILITIES

3.1 Stakeholders

This section defines the main stakeholders and clarifies their roles and responsibilities towards the detailed technical specifications of ECRIS. The following stakeholders are identified:

- For **the 27 EU Member States**:
  - The Member State delegations which are gathering in the Working Party on Cooperation in Criminal Matters (referred to as "COPEN Working Party")
  - The central authority of every Member State ensuring the implementation of the ECRIS software and its daily operation
  - The designated technical experts which gather regularly in technical workgroups, as described in the "Inception Report" document
- A **central entity** dealing with general support, technical assistance as well as dealing with all horizontal tasks relating to maintenance, evolution and dissemination of information.

3.2 Roles and Responsibilities

3.2.1 Member States

The Member States are engaged in three different ways:

1. The appointed national representatives reconvene in the COPEN Working Group which acts as the appropriate framework structure formally responsible for adoption of **decisions**. It plays the role of "Steering Committee", deciding in particular on:
   - the implementation of appropriate corrective actions in the detailed technical specifications, required for solving problematic situations identified during the development or maintenance of the national ECRIS software
   - the evolution of the detailed technical specifications
2. The central authority of every Member State is in charge of operational tasks, such as the maintenance, development and administration of the national implementation of the ECRIS software. Therefore it has the responsibility to perform the following:
   - providing the updated *service contract* to the central entity, in which the required additional information such as the actual physical address of the service endpoints has been completed
   - ensuring the correct implementation of the *service contract*
   - providing information to the central entity regarding the binary attachments implementation that is currently supported
   - ensuring that the national ECRIS software is available and able to participate appropriately in the message exchanges with other Member States

   Please note that each Member State is free to choose how these tasks are fulfilled. The role does not explicitly imply that a central authority has to perform these tasks either internally or by outsourcing them. This role merely defines the zone of responsibility of the central authority.
3. The appointed national technical experts meet on a regular basis in the technical ECRIS workgroups. They **evaluate** the possibility to apply changes, that are required either for solving problematic situations or for implementing changes requested by the judicial workgroup, to the detailed technical specifications. They act as an **advisory** board on the IT-technical aspects of the ECRIS data exchanges and provide thus support to the decision-making activity of the COPEN Working Party.

   The meetings of the ECRS technical workgroup are chaired by the specific Member State ensuring the Presidency of the Council of the European Union. The appointed technical experts may be assisted by national legal experts and vice versa since technical and legal issues are not always separate from each other.

It is recommended that the ECRIS technical workgroups meet **at least twice a year**, following the same frequency and combined with the ECRIS judicial workgroups. However these workgroups need to adopt a large degree of flexibility and may reconvene as often as required - especially if urgent matters arise.

3.2.2 Central Entity

In accordance with the decentralised nature of the ECRIS data exchanges, with the findings of the preliminary analyses and other deliverables of the *ECRIS Technical Specifications* project, a central entity is deemed necessary for effectively coordinating the works, providing organisational support, performing some technical maintenance and updates of the detailed specifications and disseminating information. The central entity can only perform changes to the detailed technical specifications if it is explicitly mandated by the COPEN Working Party to do so.

This central entity is responsible for the following activities:

- organising and coordinating the technical workgroup meetings and workshops; agenda items of these workgroups can be proposed by participating Member States; to that end, the central entity notifies sufficiently early the Member States which can have the opportunity to propose agenda items in a timely manner
- implementing the changes decided in the COPEN Working Group meetings in those technical artefacts which have been handed over to the central entity by the COPEN, and transmitting them to the national technical experts for revision
- under the mandate of the COPEN, keeping the documentation of the *ECRIS Technical Specifications* up-to-date and especially updating them in accordance with the decisions taken in the COPEN Working Group meetings
- circulating the updated versions of the technical artefacts which have been handed over to the central entity by the COPEN and related documentation to the appointed experts of the Member States
- receiving, validating and publishing the national versions of the *service contract,* the HTTPS security certificates as well as information regarding the implementation of binary exchanges and versions supported by the central authorities of the Member States

Please note that in regards to circulation of information, the central entity is free to choose the means considered most appropriate for the task, keeping though always in mind that these technical means must not create technical dependencies with the national ECRIS software implementations. In particular, central repositories such as UDDI registries have been considered and already discarded in the "Technical Architecture" document due to the fact that they would create a single point of failure for the automated ECRIS data exchanges.

According to article 3(7) of the Council Decision 2009/316/JHA, the **European Commission** "… shall provide general support and technical assistance, including the collection and drawing up of statistics …" and shall thus act as central entity for the tasks listed above that are considered as being part of general support and technical assistance. In addition, the European Commission has the mandate of producing the *ECRIS Reference Implementation* and of operating and maintaining the sTESTA network. For the other tasks to be carried out at a central level but that do not fall into the responsibility of the European Commission, the COPEN Working Party needs to mandate the appropriate body.

Please note that establishing the detailed description of the mission of the central entity is essential for a successful implementation and functioning of ECRIS but is out of scope of the *ECRIS Technical Specifications* project and is not included in this document.

4 DETAILED TECHNICAL SPECIFICATIONS

The following sections provide an in-depth presentation of the practices, approaches and solutions that are used in the technical artefacts so as to ensure interoperability and reduce information redundancy. Samples and diagrams are provided where necessary so as to illustrate the concepts being described.

4.1 General Information

The following sub-chapters provide general information about the technical artefacts that constitute the detailed technical specifications of ECRIS.

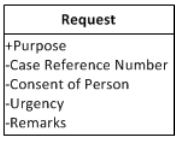
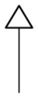
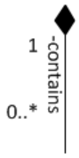
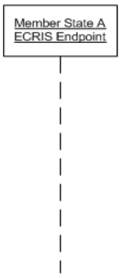
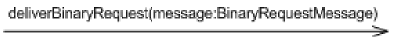
4.1.1 UML Diagrams

The diagrams in the next chapters use the UML formalism but have been simplified in order to facilitate the reading. In particular, the properties of the entities are not always shown in all diagrams.

Two types of diagrams are used:

- The class diagrams illustrate the definitions provided in the technical artefacts in terms of information entities and messages
- The sequence diagrams illustrate the kinematics (i.e. sequence of operations) to be performed, using the web service operations defined in the *WSDL* file

The following legend applies to these diagrams:

SYMBOL	DESCRIPTION
 Request +Purpose -Case Reference Number -Consent of Person -Urgency -Remarks	This symbol represents an information entity. Within the box, the simple properties contained in the entity are listed. The "+" sign indicates a mandatory property while the "-" sign indicates an optional property. For entities that are derived from their parent entity by extension rather than by restriction, the properties of the parent are not repeated in the definition.
	This type of arrow indicates that an entity extends or specialises another entity. The arrow head points to the general or abstract entity that is being extended. When used for entities that derive from a parent entity by restriction, an additional caption "restricts" is added for clearly marking the difference.
	This type of arrow indicates that an entity is composed of/contains another entity. The arrow head points to the entity being the container of the entity at the other end.
	This symbol represents an instance of an object. In the context of this document, the name refers to an instance of an ECRIS web service endpoint as a whole.
	This arrow represents an asynchronous method call with no return value. The name of the operation is stated on top of

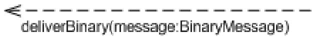
	the arrow, with the name and type of the parameter being mentioned between parentheses.
	This symbol represents an asynchronous return operation. The name of the operation is stated below the arrow, with the name and type of the parameter being mentioned between parentheses.

Table 1 – Legend for UML class diagrams

4.1.2 «Dummy» Values for Mandatory Elements

As defined in the "Business Analysis" document, each mandatory element being defined in the XSD files needs to support a "dummy" value which carries the meaning that the information is unavailable to the sender of the message.

For each such element, the XSD file defines an appropriate default value.

By convention, the following values are systematically used as defaults, depending on the technical type of the element:

- Value "UNKNOWN" for textual elements based on `xsd:string` (such as `commons:StringType`, `commons:MultilingualTextType` or `dm:YesNoUnknownStringEnumerationType`)
- Value "1800-01-01" for date elements based on `xsd:date` (i.e. `commons:StrictDateType`)

The following specific situations are however to be noted:

- The value "application/nist" has been set as default value for the MIME type of binary attachments (i.e. element `msg:BinaryAttachmentType:BinaryAttachmentMimeType`)
- The value "0" (i.e. means "unknown") has been defined for the `dm:PersonType:PersonSex` element
- For data elements defining a reference to a value contained in common reference table (such as `dm:PlaceType:PlaceCountryReference`), the value has been set to the technical identifier of the entry in the corresponding table that has the meaning "unknown" (example: value "CO-00-999-XXX" for a reference to a country)

Please note also that for very specific mandatory fields such default values are not defined purposefully. Indeed, such fields must be filled with a proper value otherwise the ECRIS data exchanges cannot take place from a technical standpoint. This concerns mainly the technical message meta-data fields such as message ID's, timestamps, correlation information, etc.

4.1.3 Transactions

The term "transaction" used throughout this document is to be understood as a sequence of individual ECRIS messages, functionally interrelated into a dialogue between two Member States. Such a dialogue is always characterised by a starting and an ending point. In particular, the word "transaction" in this document is not to be confused with "database transactions" where all actions are technically related and undone in case of an error in the execution.

The following sequence of messages illustrates a transaction as understood in this document:

The **starting point** consists of a request message sent by Member State "A" to Member State "B". "B" sends back the request deadline and then sends a "request additional ID info" message. "A" replies to this by sending back the requested identification information through an "additional ID info" message. "B" then sends the request response to "A". Because of an error in the request response, "B" cancels the previously transmitted request response message and sends a new one using corrected information. This latter response request message is the **ending point** of this transaction.

An ECRIS transaction thus **starts** when any of the following actions are occurring:

- a valid request message is received
- a valid notification message is received

According to the definitions provided in this document, including the cancellation mechanism defined later, an ECRIS transaction is to be considered as **finished** for both sending and receiving Member States when one of the following events are occurring:

- the legal deadline for the request response has passed
- the operational deadline for the notification response has passed
- a valid cancellation message is received as a response to a previously sent request
- a valid cancellation message is received as a response to previously sent notification
- a valid functional error is received as a response to a previously sent request
- a valid functional error is received as a response to a previously sent notification

4.2 Packages and Type Definitions

To avoid information redundancy and allow flexibility and extensibility, abstractions are used as much as possible in the design of the domain model. The principles of encapsulation, inheritance and polymorphism, as described in the "Technical Architecture" document, are applied to the maximum possible extent.

The following chapters provide an explanation on how the aforementioned principles are applied. The separation of the chapters is based on the packaging.

4.2.1 Commons

A common problem when creating type definitions and separating them into packages is that the dependencies created between those packages can result in circular dependencies. For example, if type definition "AType" in package "A" depends on "BType" of package "B", the dependency is considered one-way, since package "B" does not depend on package "A". If now a new type definition "CType" is added in package "B", which depends on "AType", a circular dependency between the two packages is created. Generally speaking, such circular dependencies are considered an anti-pattern as they introduce various weaknesses to the model. Concretely, such interdependencies create a tight coupling that reduces or makes impossible re-using one of the packages without the other. Furthermore, this kind of coupling can introduce a "domino" effect, where a small change in one package can cause major reworking in the depending packages.

In order to avoid such circular dependencies, it is common practice to introduce a separate package which contains definitions that are used in more than one other package. In the detailed technical specifications, this package is named "commons" and the definitions created therein are placed in the namespace "**`http://ec.europa.eu/ECRIS/commons-v1.0`**".
Additionally, with a careful selection of the type definitions included in the "commons" package, the "domino" effect mentioned above as a negative side-effect is turned into a positive feature. This is achieved by placing in the "commons" package proxies for the *XSD* base simple types, such as `xsd:date` or `xsd:string`, which are used throughout the packages. In doing so, if a structural change is required in such a base type there is no need to duplicate the change to all packages using it.

Please note that for future evolutions, the rule that should be followed in order to keep the "commons" package consistent is the following:

- Before adding a new type in the "commons" package, the following two checks must be performed and have a positive outcome
  - The type is used by at least two packages.
  - The type does not represent a specific business concept. Indeed, if a business-related type has to be re-used, the implementers should rather consider defining a new, separate package which represents the business domain this type belongs to.

The following UML class diagram provides an overview of **all** the types that are currently defined in the "commons" package:

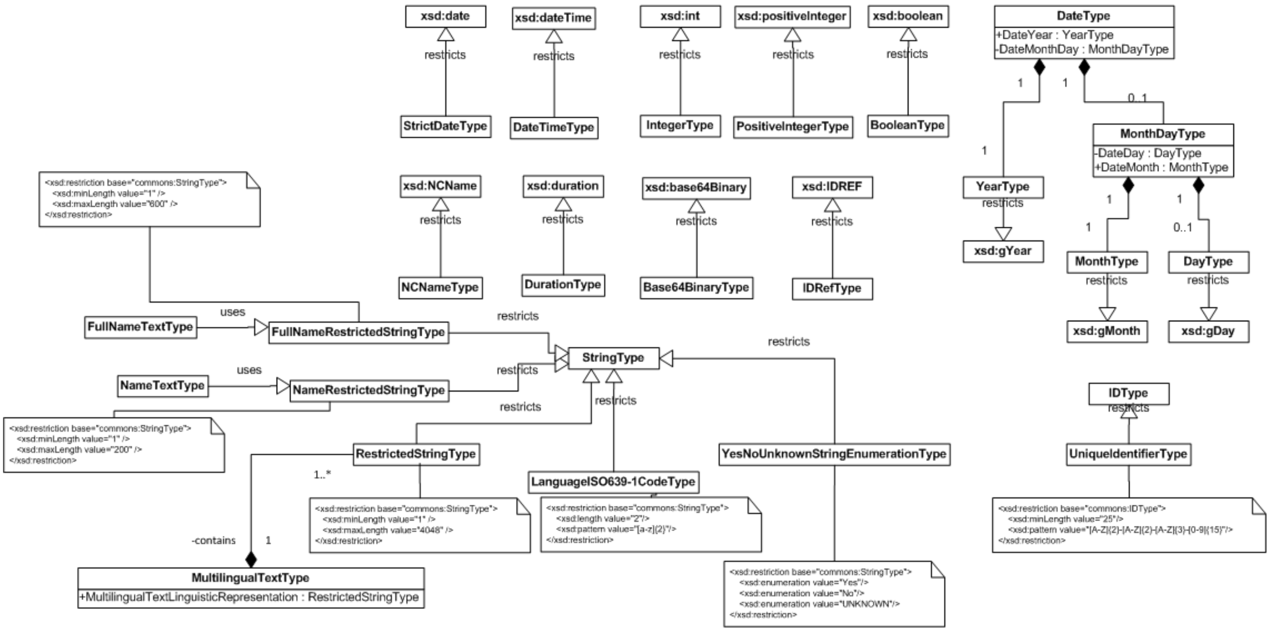


Figure 1 – Types defined in the "commons" package

Please note that most of the types contained in this package are actually "proxy" types deriving directly from native XSD types. It is to be noted though that a specific "DateType" type has been defined so as to allow defining dates that are only partially complete (i.e. year and month without day), whereas the "StrictDateType" only allows complete dates to be provided. The "DateType" type is typically used for dates that are defined in the "Business Analysis" as allowing incomplete values, such as for instance the start date of an offence. The free-text elements have been limited to 4048 characters in length by defining the "RestrictedStringType" element which is used as basis for the "MultilingualTextType" element. In addition, elements holding person names have been further limited to 200 and 600 characters in length by defining the "NameTextType" respectively the "FullNameTextType" elements that are used as basis in the other packages.

4.2.2 Common Reference Tables

The "common reference tables" package contains the structural **definition** of **all** the common reference tables used in ECRIS. The **content** of these reference tables is provided separately in one single *XML* file. This separation between structural definition and content has been chosen so as to allow easily changing the content of the common reference tables without need to modify the XSD definition of it. It is indeed expected that the XML file will be regularly modified in the scope of the regular maintenance of the ECRIS software.

The namespace defined for the "common reference table" package is "`http://ec.europa.eu/ECRIS/common-reference-tables-v1.0`".

Usage of `xsd:ID` and `xsd:IDREF`

One of the main difficulties when designing the *XSD* structure supporting the *XML* representation of the common reference tables is to keep on one hand the main principles of reusability and extensibility while on the other hand ensuring **at design time**, **by the structure** that the current but also future versions of common reference tables contain valid information on a technical level. Concretely, this mechanism ensures that the technical identifiers used within those tables are unique and when used for cross-referencing (such as cities referencing countries) the linkage between such entities is also using valid identifiers. Ensuring the uniqueness and validity of such technical identifiers indeed ensures on one hand the validity of the common reference values but also avoids data corruption when referencing to these reference values in other entities of the domain model.

The mechanism described above is implemented in the XSD definitions as follows:

- The technical identifiers of the entities defined in the *XSD* schema extend the `commons:IDType` which extends the `xsd:ID` type. This ensures the uniqueness of technical identifiers when used within the same *XML* document.

  As a consequence, if an implementer adds values to one of the common reference tables with an *XML* authoring tool that performs validation, this authoring tool will notify immediately of the error if the implementer tries to reuse an already defined technical identifier in the XML by mistake.
- In order to ensure that referencing between two entities is done with an existing technical identifier, an additional reference type is created by extending the `commons:IDRefType` which in turns extends the `xsd:IDREF` type.

  In the same way, if an implementer adds values to one of the common reference tables with an *XML* authoring tool that performs validation, this authoring tool will also notify immediately of the error if the implementer tries to establish by mistake a relation between an entity with another entity of the wrong type or using a wrong reference (for example when referencing a country in the common reference table listing cities).

Please note at this stage that such validation **at design time** can **only** work when using an appropriate *XML* authoring tool that properly implements the derived data types as specified in "XML Schema Part 2: Datatypes – Second Edition" of the *XSD* 1.0 specification, and in particular the "xsd:ID" and "xsd:IDREF" types. A few examples of such *XML* authoring tools are Altova XMLSpy, XMLNotepad and Oxygen XML.

In addition, please note that this mechanism of validation of proper and unique technical references yields correct results **only if** both the technical identifier and the reference are declared in the same *XML* document. Unfortunately this is due to the fact that the *XML* specification does not provide a way to import external documents when authoring or when validating an *XML* document. Thus, if the technical identifier used by a reference is not available to the parser within the same *XML* document, the document will be considered invalid by this parser. For this specific reason the content of all common reference tables are provided in a **single *XML* document**.

Furthermore, the approach followed in order to create the technical identifiers used in the common reference tables has to ensure that on one hand each identifier is unique and on the other hand has to facilitate tracking changes in existing values. Taking that under consideration, the general logic that is used for creating technical identifiers is the following:

[A-Z]{1,3}-[0-9]{2}-*

where

- **[A-Z]{1,3}** is a string value of 1 to 3 characters derived from the name of the specific entity for which the technical identifier is created for
- **[0-9]{2}** is a two digit number that should be incremented by 1 when a new version of an existing value is added as a new entry in the reference table
- ***** are various other entity specific elements that can ensure the uniqueness of the technical identifier

As a concrete example, the technical identifier for offence categories uses the following restriction

O-[0-9]{2}-[0-9]{6}

where

- **O** is the first letter of the word "**O**ffence"
- **[0-9]{2}** is the aforementioned versioning number. For example, the categories of offences currently defined in the common reference table all have this part set to "00". If in the future one of the existing categories needs to be redefined with a slightly different judicial meaning, a new entry can be added with the same category code but incrementing this part of the identifier to "01". This allows having a kind of logical versioning when defining new entries in the common reference tables.

- **[0-9]{6}** is a 6 digit number depicting the offence category code as it is defined in Annex A of the Council Decision 2009/316/JHA

Usage of an external reference type extending `xsd:string`

The main role of the common reference tables is to provide a standardised and codified representation of all values commonly understood in the same way by the Member States and to be used in the ECRIS data exchanges.

Performing this task requires using the technical identifiers defined for the records in the common reference tables within the body of the *XML* messages that are sent through the *web service* operations. However reusing the reference types (extending `xsd:IDREF`) or the technical identifiers themselves (extending `xsd:ID`) does not work for the same reasons as mentioned above. Indeed, using an extension of `xsd:IDREF` in the *XML* body of a message would require that the technical identifier being referenced is also declared in the same *XML* document. This would imply duplicating the definition of the common reference value being used inside the same *XML* document. Using an extension of `xsd:ID` in the *XML* body of a message would also not work since a value of that type can occur only once within an XML document.

The approach that has been chosen is to define specific element types used for the referencing of the common reference values in the *XML* body of the ECRIS messages, based on the `commons:StringType` element which is a simple text element on which using simple text elements (of types extending `commons:StringType`) on which the same restrictions of format and content are applied than for the technical identifier itself.

While this solution solves the referencing of common reference values in the *XML* body of messages without requiring duplicating the content of the common reference tables into each message, it also has the following drawbacks:

- When validating the XML message, the parser will not be able to automatically verify that a value used as a reference indeed matches an existing technical identifier in the common reference tables. Verifying that a reference points to a proper record in the expected common reference table needs thus to be done programmatically in a second step during the functional validation of the message.
- In addition, the same definition of restriction of format needs to be included up to 3 times in the XSD definition of the types: once for defining the technical identifier itself (based on `commons:IDType` → `xsd:ID`), once for defining the internal reference to the technical identifier (based on `commons:IDRefType` → `xsd:IDREF`) and once more for defining the external reference to the technical identifier (based on `commons:StringType` → `xsd:string`).

The following UML class diagram illustrates the definition of the two common reference tables' entities "CityType" and "CountryType":

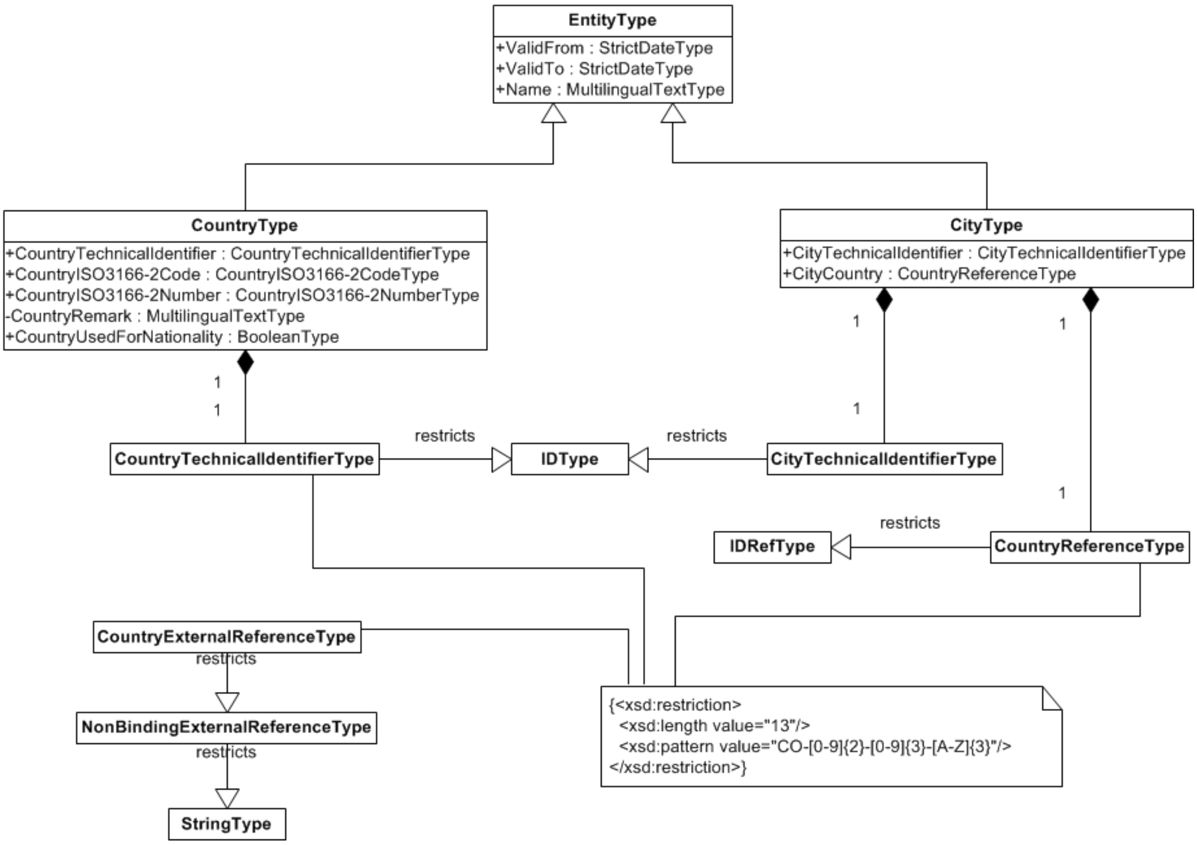

EntityType
+ValidFrom : StrictDateType
+ValidTo : StrictDateType
+Name : MultilingualTextType
CountryType
+CountryTechnicalIdentifier : CountryTechnicalIdentifierType
+CountryISO3166-2Code : CountryISO3166-2CodeType
+CountryISO3166-2Number : CountryISO3166-2NumberType
-CountryRemark : MultilingualTextType
+CountryUsedForNationality : BooleanType
CityType
+CityTechnicalIdentifier : CityTechnicalIdentifierType
+CityCountry : CountryReferenceType
1
1
1
1
1
1
CountryTechnicalIdentifierType
restricts
IDType
restricts
CityTechnicalIdentifierType
IDRefType
restricts
CountryReferenceType
CountryExternalReferenceType
restricts
NonBindingExternalReferenceType
restricts
StringType
{<xsd:restriction>
<xsd:length value="13"/>
<xsd:pattern value="CO-[0-9]{2}-[0-9]{3}-[A-Z]{3}"/>
</xsd:restriction>}

Figure 2 – Definition of "CountryType" and "CityType" common reference entities

List of common reference tables provided

Concretely, the following definitions for common reference tables are provided:

- **CentralAuthorities:** table defining the official central authorities of the Member States
- **Languages:** table defining the official languages of the Member States
- **Countries:** table defining **all** countries based on the ISO 3166-1 standard (using the 2-letter code and 3-digit numeric representation)

  (see http://en.wikipedia.org/wiki/ISO_3166-1)
- **Currencies:** table defining **all** national currencies based on the ISO 4217 standard

  (see http://en.wikipedia.org/wiki/ISO_4217)
- **CountrySubdivisions:** table defining the country subdivisions (i.e. provinces, counties, districts, etc.) of the Member States, based on the ISO 3166-2 standard

  (see http://en.wikipedia.org/wiki/ISO_3166-2)
- **Cities:** table defining the biggest cities in the 27 Member States

  Please note that this table should be expanded over time so as to add the city names most frequently used in the ECRIS data exchanges.
- **OffenceCategories:** table providing the categories of offences as defined in Annex A of the Council Decision 2009/316/JHA
- **OffenceLevelsOfParticipation:** table providing the categories for the level of participation in the offence as defined in the parameters in Annex A of the Council Decision 2009/316/JHA
- **OffenceLevelsOfCompletion:** table providing the categories for the level of completion of the offence as defined in the parameters in Annex A of the Council Decision 2009/316/JHA
- **SanctionCategories:** table providing the categories of the sanctions as defined in Annex B of the Council Decision 2009/316/JHA

- **SanctionTypesOfSuspension:** table providing the categories for the type of suspension of the sanction as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **SanctionNatures:** table providing the categories for the type of the sanction (i.e. penalty or measure) as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **SanctionAlternativeTypes:** table providing the types of alternative sanctions as defined in the parameters in Annex B of the Council Decision 2009/316/JHA
- **RequestPurposeCategories:** table providing the common categories of purposes of requests as defined in the "Business Analysis" document
- **IdentificationDocumentCategories:** table providing the common categories of identification documents used in the Member States, as defined in the "Business Analysis" document
- **DecisionChangeTypes:** table providing the types of changes that can affect a conviction as a whole or one of the sanctions within the conviction, as defined by the parameters in Annex B of the Council Decision 2009/316/JHA.
- **RequestResponses:** table providing the possible types of responses that can be sent for requests

  Please note that it contains values for positive (i.e. the request could be processed) and negative responses (i.e. the request denial and request problem messages defined in the "Business Analysis" document).
- **NotificationResponses:** table providing the possible types of responses that can be sent for notifications

  Please note that it contains values for positive (i.e. the notification receipt) and negative responses (i.e. the notification problem message defined in the "Business Analysis" document).
- **RequestAdditionalInfoResponses:** table providing the possible types of responses that can be sent for the request for additional identification information

  Please note that it contains values for positive (i.e. the additional info available) and negative responses (i.e. the additional info is not available).
- **FunctionalErrors:** table providing the list of functional error codes that can possibly occur during the functional validation of the ECRIS messages
- **Faults:** table providing the list of the supported SOAP fault codes with their descriptions
- **SanctionToSanctionsRelationshipTypes:** table providing the list of the supported relationship types for the sanction-to-sanction relations (i.e. the "SanctionToSanctionsRelationshipType" definition)
- **RequestingAuthorityTypes:** table providing the list of supported types for requesting authorities

4.2.3 Domain Model

Most of the structured information entities defined in the "Business Analysis" document are contained in the "domain model" package, with the exception of the messages and entities related conceptually to other packages such as the "Contact Person" (which is contained in the "messages" package).

The namespace defined for the "domain model" package is "`http://ec.europa.eu/ECRIS/domain-model-v1.0`".

The table listing the business entities defined in the "Business Analysis" and their technical counterparts is provided below. Where appropriate, the technical counterpart is prefixed with the namespace of the package in which it is defined.

Business entity	Technical counterpart
Person	PersonType

Contact Person	{http://ec.europa.eu/ECRIS/messages-v1.0}ContactPersonType
Requesting Authority	{http://ec.europa.eu/ECRIS/messages-v1.0}RequestingAuthorityType
Place	PlaceType
Address	PersonAddressType
Identification Document	IdentificationDocumentType
Alias	PersonAliasType
Conviction	ConvictionType
Decision	DecisionType
Offence	OffenceType
Sanction	SanctionType
Suspension	SanctionSuspensionType
Interruption	SanctionInterruptionType

Table 2 – Business entities and their technical counterparts

This package defines the entities by following closely the principles of extensibility and re-use of defined structures. In particular, abstract types are defined for re-using common definitions.

The decisions, offences and sanctions are structurally defined within the "ConvictionType" element. In addition, please note that the various relations between sanctions, offences and decisions are also defined within the "ConvictionType" entity, using the optional element "ConvictionRelationship" of type "AbstractRelationshipType". This approach ensures that all currently required types of relationships are supported and that adding new types of relationships in the future remains easy to implement. The concrete types of relations that can be established between the main entities above and which derive from the "AbstractRelationshipType" are the following:

- "DecisionToOffencesRelationshipType" allows establishing a relation between a decision and one or more offences
- "DecisionToSanctionsRelationshipType" allows establishing a relation between a decision and one or more sanctions
- "SanctionToOffencesRelationshipType" allows establishing a relation between a sanction and one or more offences
- "SanctionToSanctionsRelationshipType" allows establishing a relation between a sanction and one or more sanctions. Please note that this specific relation features a mandatory attribute "relationshipType" that indicates the type of the relationship by linking to an entry in the common reference table "SanctionToSanctionsRelationshipTypes"

Technical identifiers of main entities

In the implementation of the types above, it is important to notice that the elements "DecisionType", "OffenceType" and "SanctionType" contain each a technical identifier to be used for cross-referencing within the same *XML* document. Such technical identifiers **must be unique within an XML document**, but not necessarily across several ECRIS messages.

The "ConvictionType" carries a technical identifier that **must be unique across all ECRIS messages** and that can be used for relating to convictions externally in other *XML* documents.

As for the common reference tables, a technical identifier based on `commons:IDType` → `xsd:ID` is defined for the aforementioned entities. In addition, an internal reference to the technical identifier based on `commons:IDRefType` → `xsd:IDREF` is defined for "OffenceType" and "SanctionType" whereas the `crt:NonBindingExternalReferenceType` → `commons:StringType` is used for referencing entities of type "ConvictionType.

These identifiers have been defined with the following formats:

- For convictions: [A-Z]{2}-C-[0-9]{15} where the two first letters represent the country code of the Member State issuing the conviction. For example, if Greece is issuing a conviction, the identifier would look like the following:

  GR-C-554672903178462
- For offences: O-[0-9]{5} (example: O-80033)
- For sanctions: S-[0-9]{5} (example: S-71596)
- For decisions: D-[0-9]{5} (example: D-32157)

(where [0-9]{5} is a 5-digit number ranging from 00000 to 99999)

Please note also that no external reference type is defined for "OffenceType", "SanctionType" and "DecisionType" entities; indeed this is not necessary since these are always declared in the same *XML* document in an ECRIS message.

Please note that the fingerprints information has been defined on the message level in the XSD definition rather than in the "Person" entity so as to be able to properly reuse the structure.

Illustration

The following UML class diagram provides an overview of **some** of the entities defined the domain model, around the "ConvictionType" definition, so as to illustrate the general principle followed in the domain model definition:

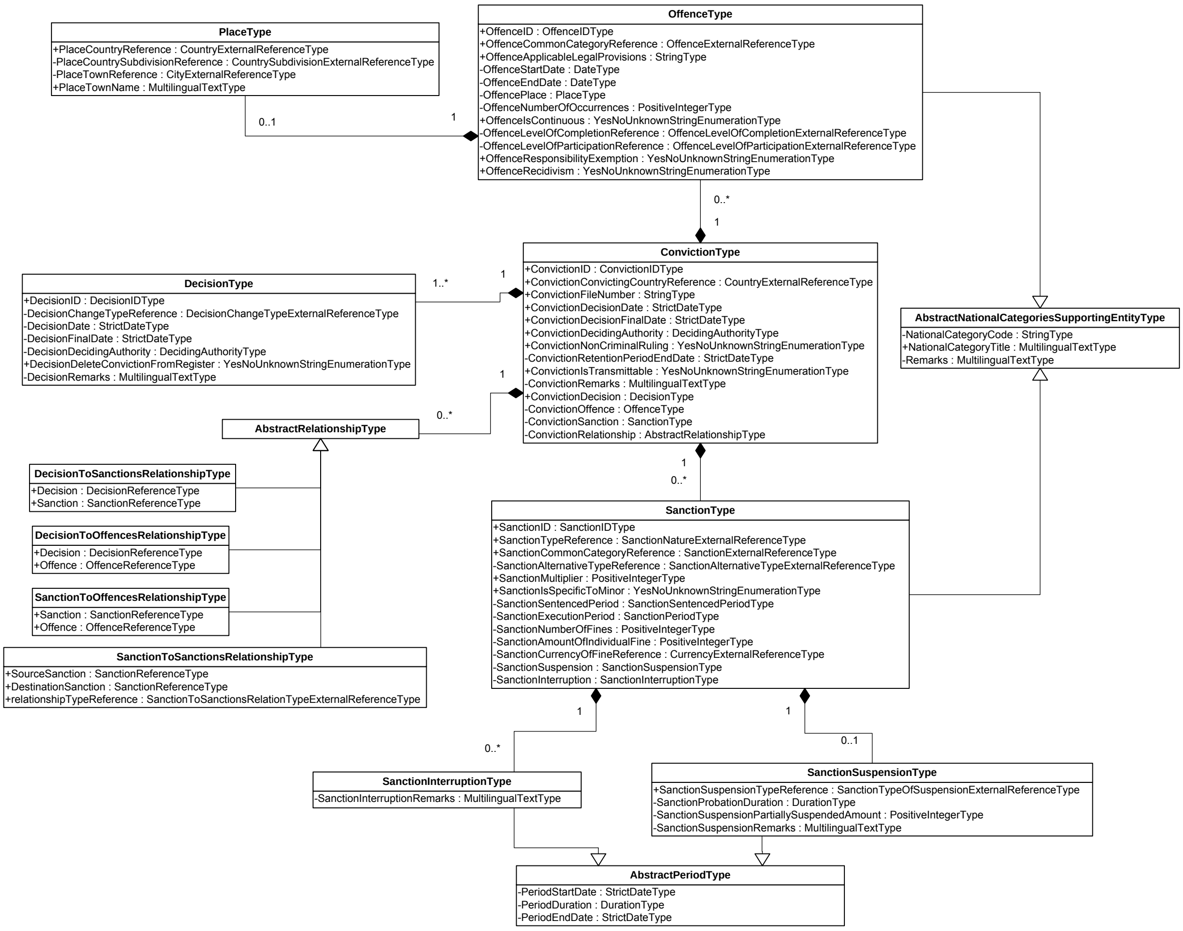

PlaceType
+PlaceCountryReference : CountryExternalReferenceType
-PlaceCountrySubdivisionReference : CountrySubdivisionExternalReferenceType
-PlaceTownReference : CityExternalReferenceType
+PlaceTownName : MultilingualTextType
0..1
1
OffenceType
+OffenceID : OffenceIDType
+OffenceCommonCategoryReference : OffenceExternalReferenceType
+OffenceApplicableLegalProvisions : StringType
-OffenceStartDate : DateType
-OffenceEndDate : DateType
-OffencePlace : PlaceType
-OffenceNumberOfOccurrences : PositiveIntegerType
+OffenceIsContinuous : YesNoUnknownStringEnumerationType
-OffenceLevelOfCompletionReference : OffenceLevelOfCompletionExternalReferenceType
-OffenceLevelOfParticipationReference : OffenceLevelOfParticipationExternalReferenceType
+OffenceResponsibilityExemption : YesNoUnknownStringEnumerationType
+OffenceRecidivism : YesNoUnknownStringEnumerationType
0..*
1
DecisionType
+DecisionID : DecisionIDType
-DecisionChangeTypeReference : DecisionChangeTypeExternalReferenceType
-DecisionDate : StrictDateType
-DecisionFinalDate : StrictDateType
-DecisionDecidingAuthority : DecidingAuthorityType
+DecisionDeleteConvictionFromRegister : YesNoUnknownStringEnumerationType
-DecisionRemarks : MultilingualTextType
1..*
1
ConvictionType
+ConvictionID : ConvictionIDType
+ConvictionConvictingCountryReference : CountryExternalReferenceType
+ConvictionFileNumber : StringType
+ConvictionDecisionDate : StrictDateType
+ConvictionDecisionFinalDate : StrictDateType
+ConvictionDecidingAuthority : DecidingAuthorityType
+ConvictionNonCriminalRuling : YesNoUnknownStringEnumerationType
-ConvictionRetentionPeriodEndDate : StrictDateType
+ConvictionIsTransmittable : YesNoUnknownStringEnumerationType
-ConvictionRemarks : MultilingualTextType
+ConvictionDecision : DecisionType
-ConvictionOffence : OffenceType
-ConvictionSanction : SanctionType
-ConvictionRelationship : AbstractRelationshipType
AbstractNationalCategoriesSupportingEntityType
-NationalCategoryCode : StringType
+NationalCategoryTitle : MultilingualTextType
-Remarks : MultilingualTextType
1
0..*
AbstractRelationshipType
DecisionToSanctionsRelationshipType
+Decision : DecisionReferenceType
+Sanction : SanctionReferenceType
DecisionToOffencesRelationshipType
+Decision : DecisionReferenceType
+Offence : OffenceReferenceType
SanctionToOffencesRelationshipType
+Sanction : SanctionReferenceType
+Offence : OffenceReferenceType
SanctionToSanctionsRelationshipType
+SourceSanction : SanctionReferenceType
+DestinationSanction : SanctionReferenceType
+relationshipTypeReference : SanctionToSanctionsRelationTypeExternalReferenceType
1
0..*
SanctionType
+SanctionID : SanctionIDType
+SanctionTypeReference : SanctionNatureExternalReferenceType
+SanctionCommonCategoryReference : SanctionExternalReferenceType
-SanctionAlternativeTypeReference : SanctionAlternativeTypeExternalReferenceType
+SanctionMultiplier : PositiveIntegerType
+SanctionIsSpecificToMinor : YesNoUnknownStringEnumerationType
-SanctionSentencedPeriod : SanctionSentencedPeriodType
-SanctionExecutionPeriod : SanctionPeriodType
-SanctionNumberOfFines : PositiveIntegerType
-SanctionAmountOfIndividualFine : PositiveIntegerType
-SanctionCurrencyOfFineReference : CurrencyExternalReferenceType
-SanctionSuspension : SanctionSuspensionType
-SanctionInterruption : SanctionInterruptionType
1
0..*
1
0..1
SanctionInterruptionType
-SanctionInterruptionRemarks : MultilingualTextType
SanctionSuspensionType
+SanctionSuspensionTypeReference : SanctionTypeOfSuspensionExternalReferenceType
-SanctionProbationDuration : DurationType
-SanctionSuspensionPartiallySuspendedAmount : PositiveIntegerType
-SanctionSuspensionRemarks : MultilingualTextType
AbstractPeriodType
-PeriodStartDate : StrictDateType
-PeriodDuration : DurationType
-PeriodEndDate : StrictDateType

Figure 3 – Domain Model – "ConvictionType" and related types

4.2.4 Messages

All messages that can be sent from one Member State to another one in the *web service* operations have been defined in the "messages" package and are placed in the namespace: **"http://ec.europa.eu/ECRIS/messages-v1.0"**.

Here also the principles of abstraction and re-use are applied: a base abstract type named "AbstractMessageType" defines the basic message structure composed of message meta-data and message data. Please note that the abstract message data type is defined as being of type "xsd:anyType". This is done for allowing defining more specific message data types by restriction. The message meta-data property is however strongly typed.

The following UML class diagram illustrates the composition and structure of the "AbstractMessageType":

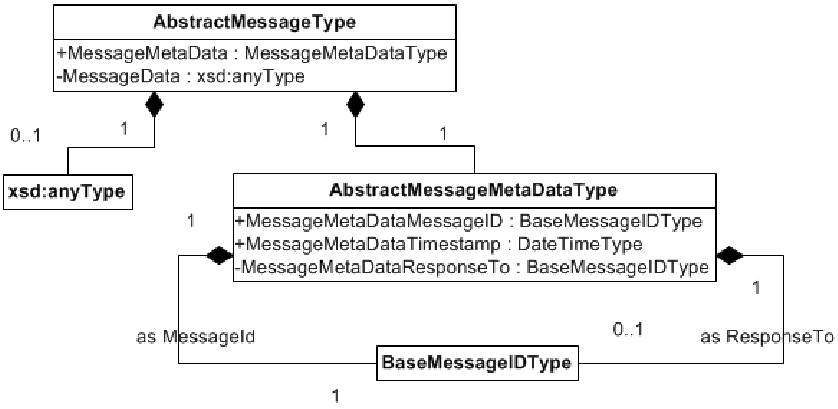

AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
0..1
1
1
1
xsd:anyType
AbstractMessageMetaDataType
+MessageMetaDataMessageID : BaseMessageIDType
+MessageMetaDataTimestamp : DateTimeType
-MessageMetaDataResponseTo : BaseMessageIDType
1
1
as MessageId
0..1
as ResponseTo
BaseMessageIDType
1

Figure 4 – Messages – "AbstractMessageType"

Following this approach, each specific message derived from this type **by restriction** must define its message data structure, if any, as well as an extension of the message meta-data type.

The following UML diagrams illustrate the definitions of the concrete messages and how they are derived from "AbstractMessageType":

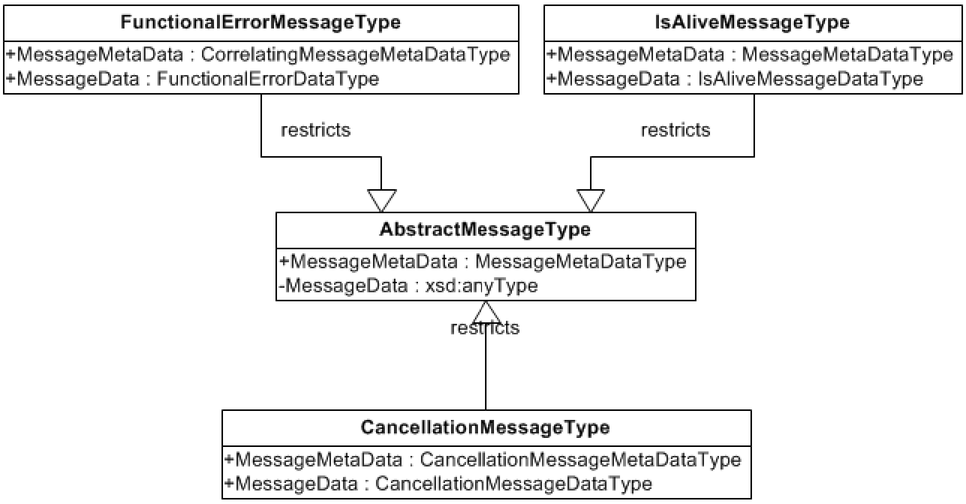

FunctionalErrorMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : FunctionalErrorDataType
IsAliveMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : IsAliveMessageDataType
restricts
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
restricts
CancellationMessageType
+MessageMetaData : CancellationMessageMetaDataType
+MessageData : CancellationMessageDataType

Figure 5 – Messages – Concrete message types for "IsAlive", "FunctionalError" and "Cancellation" implementation messages

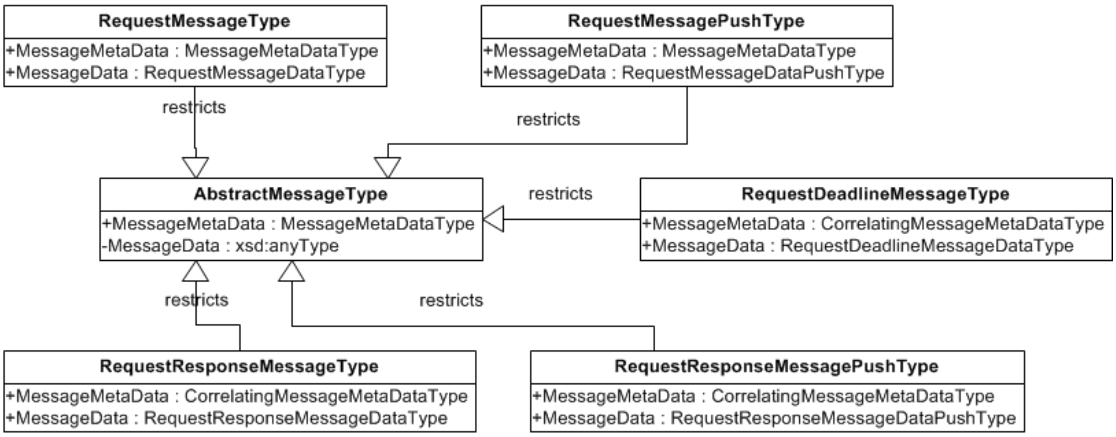

RequestMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : RequestMessageDataType
RequestMessagePushType
+MessageMetaData : MessageMetaDataType
+MessageData : RequestMessageDataPushType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
RequestDeadlineMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestDeadlineMessageDataType
RequestResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestResponseMessageDataType
RequestResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestResponseMessageDataPushType

Figure 6 – Messages – Concrete message types for the "request" business process

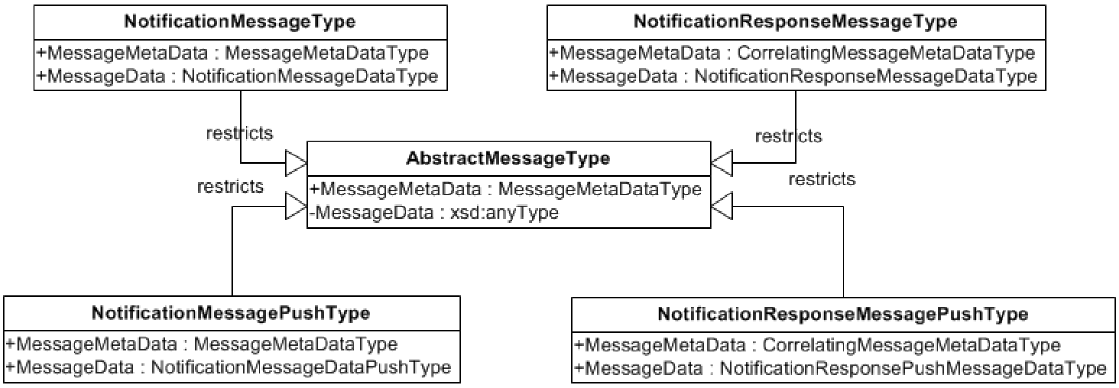

NotificationMessageType
+MessageMetaData : MessageMetaDataType
+MessageData : NotificationMessageDataType
NotificationResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : NotificationResponseMessageDataType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
NotificationMessagePushType
+MessageMetaData : MessageMetaDataType
+MessageData : NotificationMessageDataPushType
NotificationResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : NotificationResponsePushMessageDataType

Figure 7 – Messages – Concrete message types for the "notification" business process

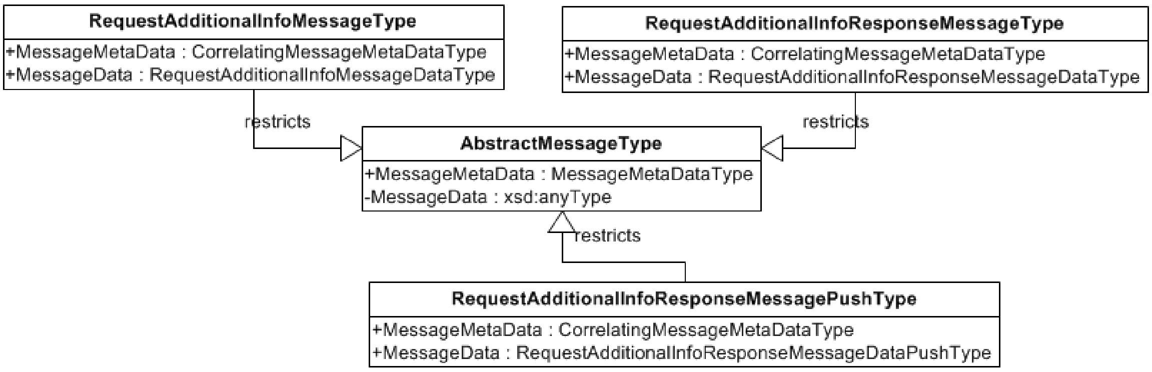

RequestAdditionalInfoMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoMessageDataType
RequestAdditionalInfoResponseMessageType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoResponseMessageDataType
restricts
AbstractMessageType
+MessageMetaData : MessageMetaDataType
-MessageData : xsd:anyType
RequestAdditionalInfoResponseMessagePushType
+MessageMetaData : CorrelatingMessageMetaDataType
+MessageData : RequestAdditionalInfoResponseMessageDataPushType

Figure 8 – Messages – Concrete messages of the "search person" sub-process

In the diagrams above, please note that each message defines its own message data type, according to the information structure required specifically by the message.

Business messages versus technical messages

As mentioned in the "Business Analysis", the business messages do not necessarily have a one to one correspondence with the messages defined in the technical *service contract.* Indeed, messages that carry identification information have to support binary attachments for fingerprints and thus require additional messages so as to implement both "Push" and "Pull" approaches as defined in the "Technical Architecture" document. Additionally, with the interest of keeping the number of message types and operations to a minimum and avoid information or structural redundancy, several operations and messages described in the "Business Analysis" are implemented using one message and operation definition.

The following table provides the messages defined in the "Business Analysis" and their technical counterparts:

Business Message	Technical Message
Notification	NotificationMessageType NotificationMessagePushType
Notification Problem	NotificationResponseMessageType
Notification Receipt	NotificationResponseMessageType NotificationResponseMessagePushType
Request	RequestMessageType RequestMessagePushType
Request Deadline	RequestDeadlineMessageType
Request Denial	RequestResponseMessageType
Request Problem	RequestResponseMessageType
Request Response	RequestResponseMessageType RequestResponseMessagePushType
Request Additional ID Info	RequestAdditionalInfoMessageType
Additional ID Info	RequestAdditionalInfoResponseMessageType RequestAdditionalInfoResponseMessagePushType
Additional ID Info Unavailable	RequestAdditionalInfoResponseMessageType

Table 3 – Business messages and their technical counterparts

The various possible responses to a notification, to a request and to a request for additional identification information have been grouped into the technical messages "NotificationResponse", "RequestResponse" and "RequestAdditionalInfoResponse". In each such technical response message, a response type indicates the nature of the response and provides the more fine-grained result depending on the type of message. The definition of the response types for each message is provided in the following common reference tables which can be expanded easily for example for adding specific causes for not providing a positive response:

- RequestResponses
- NotificationResponses
- RequestAdditionalInfoResponses

Grouping the response messages can prove useful from the point of view of implementers, especially when monitoring the transactions. Indeed, the "notification response" and "request response" messages are always closing the ongoing "notification" or "request" transaction.

For each message carrying personal identification information, a corresponding "push" message is defined. The simple message may carry only meta-data about the attached binary fingerprints file while the "push" message may carry the actual binary file. The principle is further detailed later in this document.

Message Meta-Data

As defined in the "Technical Architecture" document, the functional information is to be clearly separated from the technical information in the body of the *XML* message. To that end, a series of data structures are defined for carrying the technical meta-data of the message.

These meta-data structures are based on the "AbstractMessageMetaDataType" described above and contain the following properties:

- "MessageMetaDataMessageID" carries the technical identifier of the message. Please note that the ECRIS messages exchanged **are required** to be **uniquely identifiable** not only on a national level, but across all ECRIS message exchanges so as to be able to properly correlate them.

The technical identifier type is always mandatory. The "BaseMessageIDType", from which all other ID types are derived, defines the following format: **[A-Z]{2}-[A-Z]{2}-[A-Z]{3}-[0-9]{15}** where:

- **[A-Z]{2}** is the two letter code of the sending Member State in **capitals**
- **[A-Z]{2}** is the two letter code of the receiving Member State in **capitals**
- **[A-Z]{3}** is a 3 character code for the message type. The following values are valid
  - REQ for requests

    (types "RequestMessageType", "RequestPushMessageType")
  - RDL for request deadlines

    (type "RequestDeadlineMessageType")
  - RRS for request responses

    (types "RequestResponseMessageType", "RequestResponseMessagePushType")
  - NOT for notifications

    (types "NotificationMessageType", "NotificationMessagePushType")
  - NRS for notification responses

    (types "NotificationResponseMessageType", "NotificationResponsePushMessageType")
  - RAI for requests for additional information

    (type "RequestAdditionalInfoMessageType")
  - RAR for responses to additional information requests

    (types "RequestAdditionalInfoResponseMessageType", "RequestAdditionalInfoResponsePushMessageType")
  - IAL for "isAlive" messages
  - FEM for "FunctionalErrorMessageType"
  - CAN for "CancellationMessageType"
- **[0-9]{15}** is a number consisting of 15 digits (an implementer can freely choose to either iterate this number each time a new message is created or choose a random number from a provided pool, as long as it ensures that this number has not been used before)

Please note that the numeric part of this technical identifier is defined as being unique on a national level and the technical identifier is made unique across all ECRIS messages by prefixing the numeric part with the Member State code.

Please note also that, depending on the type of message, different combinations of ID types for the "MessageMetaDataMessageID" and "MessageMetaDataResponseTo" are defined. This is done in order to enforce the rules defined in kinematics for message correlation and allowed message exchanges. As an example, the "CancelledMessageIDType" can only accept as values ID's of notification, notification response, request and requests response messages.

As an example, if Germany intends to send a notification message to Austria using ECRIS, the technical identifier used would look like "DE-AT-NOT-153264871659485" or "DE-AT-NOT-000000000015234".

Please note also that **no dummy value** must be used in this field.

- "MessageMetaDataTimestamp" carries the date and time of the moment when the message is **sent** by the Member State. This property is always mandatory; in particular no dummy value must be used in this field.

  The following rules must be observed for filling in this field properly:

  - the date and time up to the seconds should be inserted just before sending the message

- the date/time used should be the local date/time and the indication of the time-zone must be included

The expected values are thus of the following form: 2010-11-25T21:32:35+02:00

- "MessageMetaDataResponseTo" is a property that contains the technical identifier of the message to which this message is responding to (i.e. value of "MessageMetaDataMessageID" of a previous message). It allows thus correlating a message to one previously sent.

  In the ECRIS message exchanges, the messages are to be correlated in a **sequential** manner, meaning that the latest message must reference the previously **received** message **within the same transaction**.

  As a concrete example, let's assume that Lithuania sends a request message with a technical ID "LT-PL-REQ-123432653223112" to Poland. In response to that message, Poland sends a new message "PL-LT-RDL-321234177654353" containing the legal deadline for responding to the request. This message contains the "MessageMetaDataResponseTo" property filled with the technical identifier provided in the request of Lithuania: "LT-PL-REQ-123432653223112". Let's assume now that Poland finishes successfully the process and returns a message to Lithuania containing the requested information. This response message has for instance the unique technical identifier "PL-LT-RRS-32123417765435**4**". It also contains the "MessageMetaDataResponseTo" property filled with the technical identifier of the latest **received** message within this transaction, which is thus the request message that was sent by Lithuania.

  (Please note that the numeric part of the technical message identifier is not necessarily incremented by one unit and is not necessary assigned sequentially for several messages in a transaction.)

As described above, different extensions of the "AbstractMessageMetaDataType" have been defined so as to enforce the rules foreseen for message correlation and kinematics:

- **"MessageMetaDataType":** A message meta-data type used for all messages which are entry points for business processes and which do not correlate to previously send messages**.**
- **"CorrelatingMessageMetaDataType":** An extension to the abstract meta-data type used for all messages which are not entry points for business processes and which **MUST** correlate to a previously received message.
- **"CancellationMessageMetaDataType":** A meta-data type definition, used for transmitting cancellation messages.

  Attention: in the case of a cancellation message, the "MessageMetaDataResponseTo" property is filled with the technical identifier of the message **previously sent and being cancelled** within this transaction.

As a rule, the messages that initiate a new transaction (i.e. "NotificationMessageType", "NotificationMessagePushType", "RequestMessageType" and "RequestPushMessageType") use "MessageMetaDataType" as message meta-data. All subsequent messages use the "CorrelatingMessageMetaDataType". The "CancellationMessageMetaDataType" is only used for transmitting cancellation messages and thus the property "MessageMetaDataResponseTo" allows indicating only ID's for the types of messages that can actually be cancelled: request, request response, notification and notification response messages.

For illustrating the proper usage of the technical identifiers of messages and of the "MessageMetaDataResponseTo" property, let's take a complex kinematic in which Belgium sends a request to Poland (please note that the names of properties and numeric parts of the technical identifiers have been shortened for a better readability):

Request (BE > PL)	MessageID = BE-PL-REQ-123…456

Request deadline (PL > BE)	MessageID = PL-BE-RDL-987…321
	ResponseTo = BE-PL-REQ-123…456

Request for additional ID info	MessageID = PL-BE-RAI-987…322

(PL > BE)	ResponseTo = BE-PL-REQ-123…456

Response with additional ID info (BE > PL)	MessageID = BE-PL-RAR-123…463
	ResponseTo = PL-BE-RAI-987…322

Functional error (PL > BE)	MessageID = PL-BE-FEM-987…327
	ResponseTo = BE-PL-RAR-123…463

Response with additional ID info (BE > PL)	MessageID = BE-PL-RAR-123…466
	ResponseTo = PL-BE-FEM-987…327

Request response (PL > BE)	MessageID = PL-BE-RRS-987…331
	ResponseTo = BE-PL-RAR-123…466

Cancellation of request response (PL > BE)	MessageID = PL-BE-CAN-987…336
	ResponseTo = PL-BE-RRS-987…331

Request response (PL > BE)	MessageID = PL-BE-RRS-987…339
	ResponseTo = BE-PL-RAR-123…466

The following UML class diagram provides an overview of the structures to be used as message meta-data:

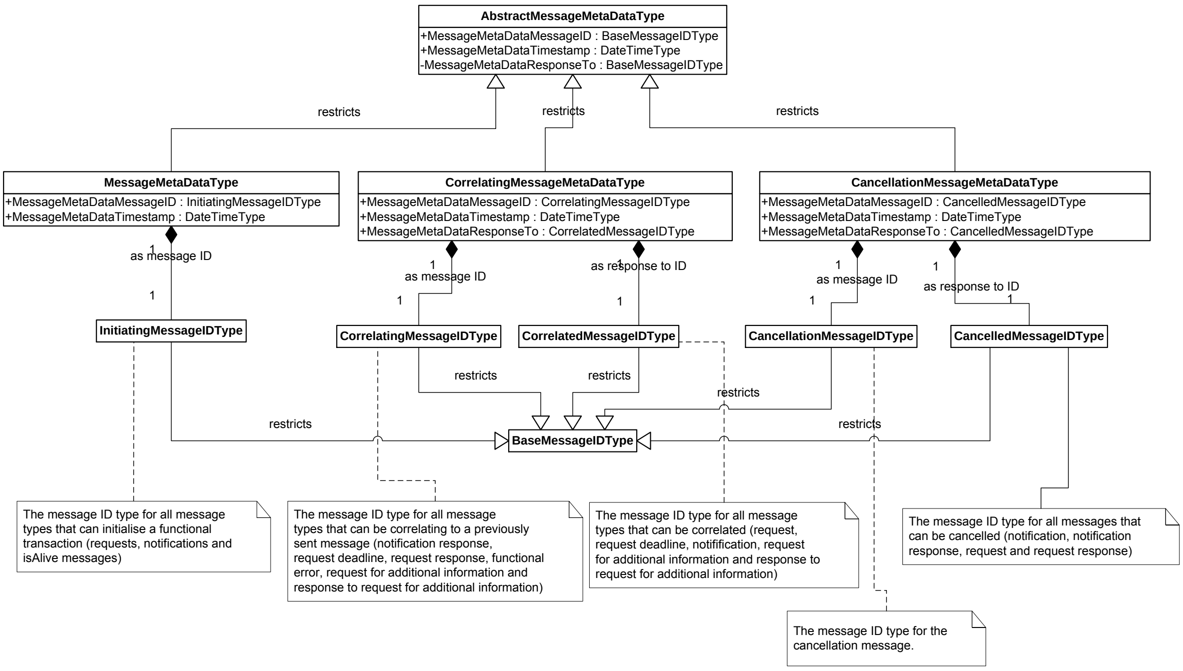

AbstractMessageMetaDataType
+MessageMetaDataMessageID : BaseMessageIDType
+MessageMetaDataTimestamp : DateTimeType
-MessageMetaDataResponseTo : BaseMessageIDType
restricts
restricts
restricts
MessageMetaDataType
+MessageMetaDataMessageID : InitiatingMessageIDType
+MessageMetaDataTimestamp : DateTimeType
CorrelatingMessageMetaDataType
+MessageMetaDataMessageID : CorrelatingMessageIDType
+MessageMetaDataTimestamp : DateTimeType
+MessageMetaDataResponseTo : CorrelatedMessageIDType
CancellationMessageMetaDataType
+MessageMetaDataMessageID : CancelledMessageIDType
+MessageMetaDataTimestamp : DateTimeType
+MessageMetaDataResponseTo : CancelledMessageIDType
1
as message ID
1
1
as message ID
1
as response to ID
1
1
1
as message ID
1
1
as response to ID
1
InitiatingMessageIDType
CorrelatingMessageIDType
CorrelatedMessageIDType
CancellationMessageIDType
CancelledMessageIDType
restricts
restricts
restricts
restricts
restricts
BaseMessageIDType
The message ID type for all message types that can initialise a functional transaction (requests, notifications and isAlive messages)
The message ID type for all message types that can be correlating to a previously sent message (notification response, request deadline, request response, functional error, request for additional information and response to request for additional information)
The message ID type for all message types that can be correlated (request, request deadline, notifification, request for additional information and response to request for additional information)
The message ID type for all messages that can be cancelled (notification, notification response, request and request response)
The message ID type for the cancellation message.

Figure 9 – Messages – Meta-Data types

Message Data

The message data part contains the functional information of the ECRIS message, and is thus clearly separated from the technical meta-data as explained previously.

The design principle that has been chosen for implementing this message data is to define a common base abstract type named "AbstractBusinessMessageDataType" and containing **all possible properties** of all business messages. All concrete message data types are then derived from this abstract type by **restriction**, only keeping the elements that are relevant to the specific message and redefining the cardinality as appropriate.

The main advantage is that the same structure and sequence of elements is enforced for all similar messages by their parent abstract type.

The following UML class diagram illustrates the "AbstractBusinessMessageDataType" element:

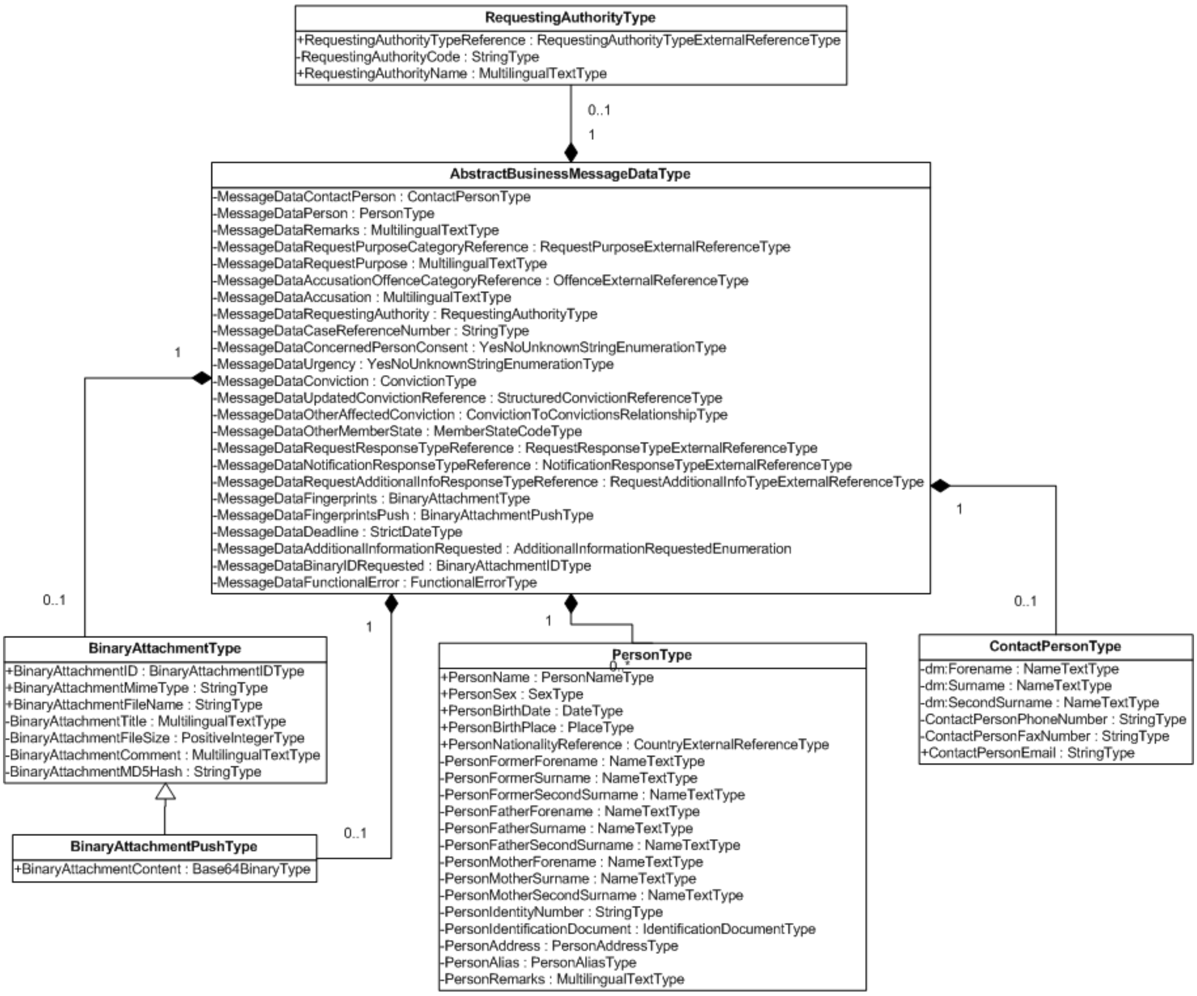

RequestingAuthorityType
+RequestingAuthorityTypeReference : RequestingAuthorityTypeExternalReferenceType
-RequestingAuthorityCode : StringType
+RequestingAuthorityName : MultilingualTextType
0..1
1
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
-MessageDataFunctionalError : FunctionalErrorType
1
0..1
1
0..1
1
1
0..*
1
0..1
BinaryAttachmentType
+BinaryAttachmentID : BinaryAttachmentIDType
+BinaryAttachmentMimeType : StringType
+BinaryAttachmentFileName : StringType
-BinaryAttachmentTitle : MultilingualTextType
-BinaryAttachmentFileSize : PositiveIntegerType
-BinaryAttachmentComment : MultilingualTextType
-BinaryAttachmentMD5Hash : StringType
BinaryAttachmentPushType
+BinaryAttachmentContent : Base64BinaryType
PersonType
+PersonName : PersonNameType
+PersonSex : SexType
+PersonBirthDate : DateType
+PersonBirthPlace : PlaceType
+PersonNationalityReference : CountryExternalReferenceType
-PersonFormerForename : NameTextType
-PersonFormerSurname : NameTextType
-PersonFormerSecondSurname : NameTextType
-PersonFatherForename : NameTextType
-PersonFatherSurname : NameTextType
-PersonFatherSecondSurname : NameTextType
-PersonMotherForename : NameTextType
-PersonMotherSurname : NameTextType
-PersonMotherSecondSurname : NameTextType
-PersonIdentityNumber : StringType
-PersonIdentificationDocument : IdentificationDocumentType
-PersonAddress : PersonAddressType
-PersonAlias : PersonAliasType
-PersonRemarks : MultilingualTextType
ContactPersonType
-dm:Forename : NameTextType
-dm:Surname : NameTextType
-dm:SecondSurname : NameTextType
-ContactPersonPhoneNumber : StringType
-ContactPersonFaxNumber : StringType
+ContactPersonEmail : StringType

Figure 10 – Messages – "AbstractBusinessMessageDataType"

The following UML class diagrams illustrate the specific concrete message data types that are derived by restriction from the "AbstractBusinessMessageDataType":

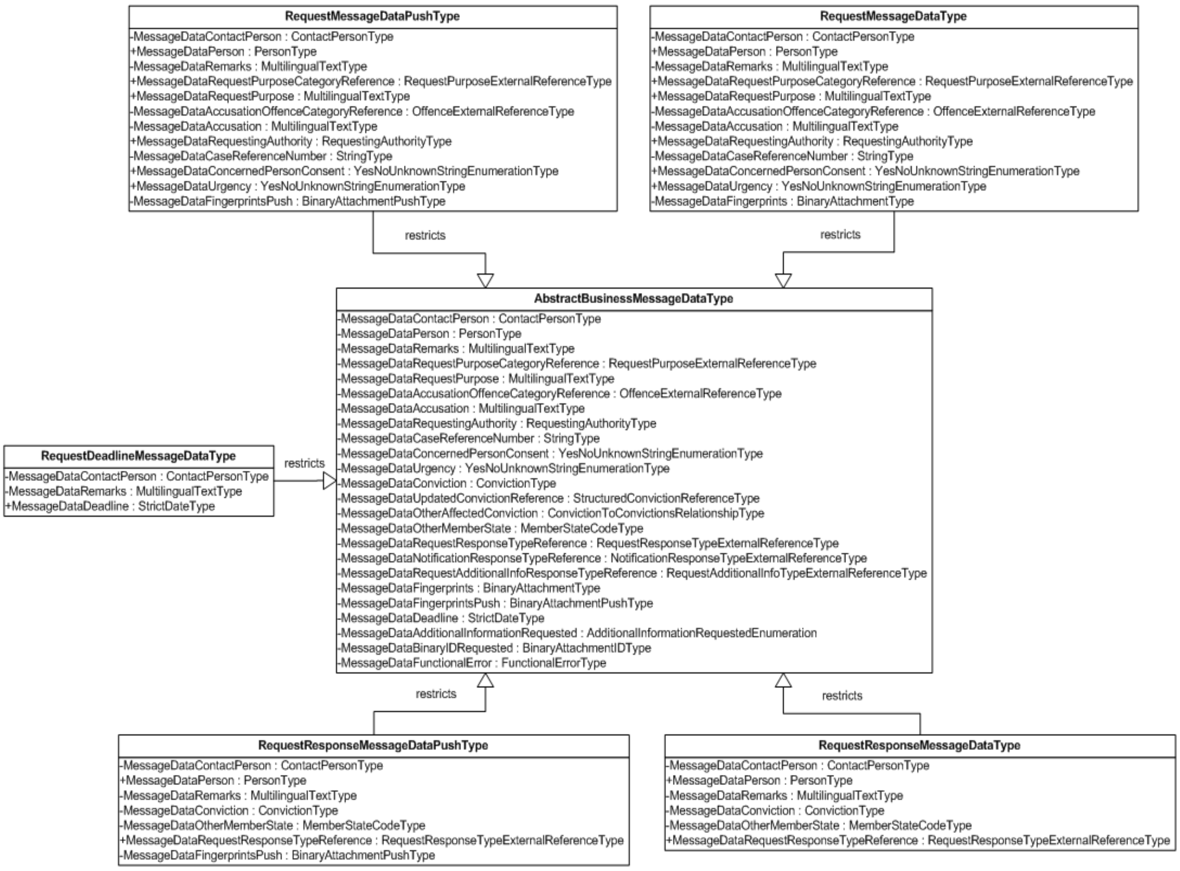

RequestMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
+MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
+MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
+MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
+MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
RequestMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
+MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
+MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
+MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
+MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataFingerprints : BinaryAttachmentType
restricts
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
-MessageDataFunctionalError : FunctionalErrorType
RequestDeadlineMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataDeadline : StrictDateType
restricts
restricts
restricts
RequestResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataConviction : ConvictionType
-MessageDataOtherMemberState : MemberStateCodeType
+MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
RequestResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataConviction : ConvictionType
-MessageDataOtherMemberState : MemberStateCodeType
+MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType

Figure 11 – Message data types for "request" business process messages

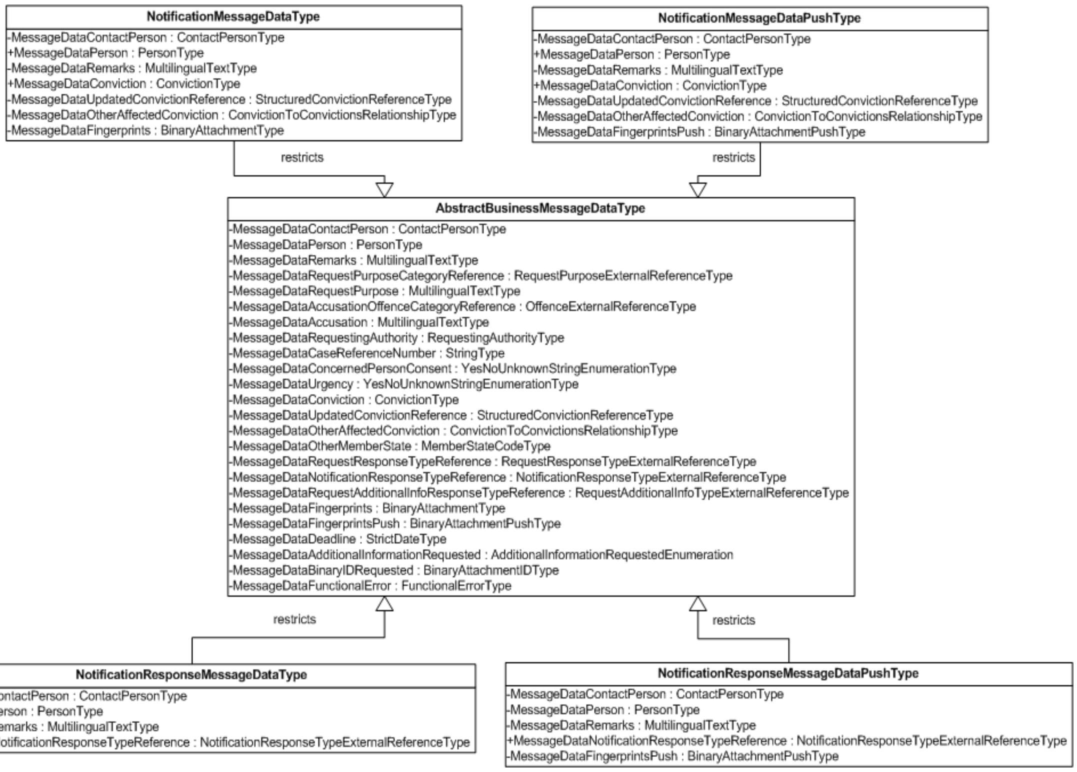

NotificationMessageDataType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataFingerprints : BinaryAttachmentType
NotificationMessageDataPushType
-MessageDataContactPerson : ContactPersonType
+MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
restricts
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
-MessageDataFunctionalError : FunctionalErrorType
restricts
restricts
NotificationResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
NotificationResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType

Figure 12 – Message data types for "notification" business process messages

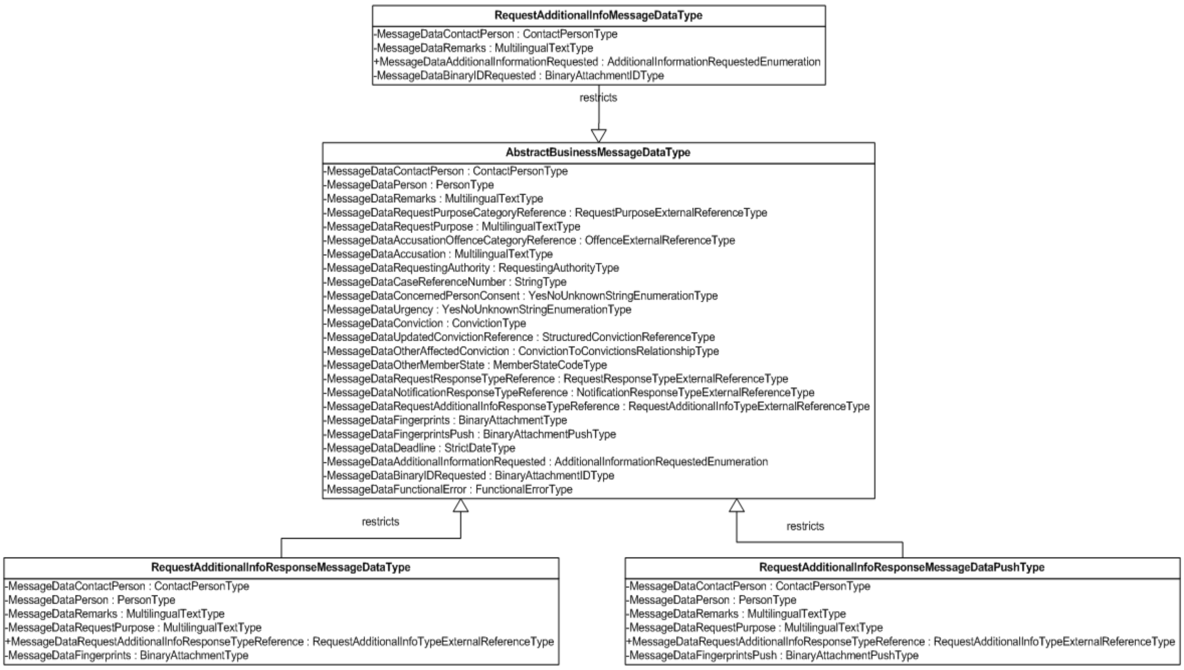

RequestAdditionalInfoMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataRemarks : MultilingualTextType
+MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
restricts
AbstractBusinessMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurposeCategoryReference : RequestPurposeExternalReferenceType
-MessageDataRequestPurpose : MultilingualTextType
-MessageDataAccusationOffenceCategoryReference : OffenceExternalReferenceType
-MessageDataAccusation : MultilingualTextType
-MessageDataRequestingAuthority : RequestingAuthorityType
-MessageDataCaseReferenceNumber : StringType
-MessageDataConcernedPersonConsent : YesNoUnknownStringEnumerationType
-MessageDataUrgency : YesNoUnknownStringEnumerationType
-MessageDataConviction : ConvictionType
-MessageDataUpdatedConvictionReference : StructuredConvictionReferenceType
-MessageDataOtherAffectedConviction : ConvictionToConvictionsRelationshipType
-MessageDataOtherMemberState : MemberStateCodeType
-MessageDataRequestResponseTypeReference : RequestResponseTypeExternalReferenceType
-MessageDataNotificationResponseTypeReference : NotificationResponseTypeExternalReferenceType
-MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
-MessageDataFingerprintsPush : BinaryAttachmentPushType
-MessageDataDeadline : StrictDateType
-MessageDataAdditionalInformationRequested : AdditionalInformationRequestedEnumeration
-MessageDataBinaryIDRequested : BinaryAttachmentIDType
-MessageDataFunctionalError : FunctionalErrorType
restricts
restricts
RequestAdditionalInfoResponseMessageDataType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurpose : MultilingualTextType
+MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprints : BinaryAttachmentType
RequestAdditionalInfoResponseMessageDataPushType
-MessageDataContactPerson : ContactPersonType
-MessageDataPerson : PersonType
-MessageDataRemarks : MultilingualTextType
-MessageDataRequestPurpose : MultilingualTextType
+MessageDataRequestAdditionalInfoResponseTypeReference : RequestAdditionalInfoTypeExternalReferenceType
-MessageDataFingerprintsPush : BinaryAttachmentPushType

Figure 13 – Message data types for "search person" business sub-process messages

Please note that the property "MessageDataUrgency" is defined as being of type "YesNoUnknownStringEnumerationType". The values of this specific element must thus be interpreted as follows:

- "Yes" corresponds to a "High" urgency
- "No" corresponds to a "Normal" urgency
- "UNKNOWN" indicates that no value is provided for the urgency of the request

4.3 *Web Service* Operations

4.3.1 Style and Setting

The *web service* defined in the *WSDL* file follows the "Document" style using "Literal" as the "use" setting for the "soap:body" element. This style and setting is preferred to "RPC" style and "Encoding" setting mainly due to the fact that only this approach fulfils the requirements defined in the "Technical Architecture" document for performing synchronous structural validation of the message. Additionally using this style and setting ensures that the footprint of messages exchanged remains small because no encoding information is added in the message. Please note also that the *WS-I Basic Profile 1.1* specification, although not directly supported by the ECRIS detailed technical specifications, encourages the use of "Document" style using "Literal" setting and hints to the fact that the combination of "RPC" style with "Encoding" setting will most probably become deprecated in the future.

4.3.2 Transmission Primitive

Furthermore, the transmission primitive chosen for the operations (with the exception of the "IsAlive" operation) is the "one-way" primitive as defined in the *WSDL* 1.1 specification in chapter "2.4.1 – One-way Operation" (http://www.w3.org/TR/wsdl#_one-way). This transmission primitive is the most appropriate one given the fact that all operations (with the exception of "IsAlive") do not return a synchronous response message.

The main drawback of this primitive though is that, by definition, it does not foresee faults. However, since the definition of the *WSDL* specification, many implementers have realised that it is frequently necessary to be able to return faults in cases of errors, also when using operations that do not return response messages. Therefore, although not explicit in the *WSDL* 1.1 specification, most *web service* frameworks support returning faults when performing operations with input only parameters. Also, please note that in *WSDL* 2.0, which is the succeeding specification of WSDL 1.1, an appropriate transmission primitive named "robust-in-only" has been foreseen in order to accommodate such situations (see also http://www.w3.org/TR/wsdl20-adjuncts/#robust-in-only).

The solution adopted in the technical specifications is to define *web service* operations **without** faults on the *service contract* level. However, if a technical error occurs, implementers **must** return the standard "*SOAP* Fault" element and use either the standard fault codes or the custom fault codes that are defined in the common reference table "Faults". This way of working has been thoroughly tested with various technical platforms and is generally very well supported.

Please note that another alternative was initially considered and tested. Indeed the initial approach was to define custom fault types as part of the domain model and declare them accordingly in the *service contract* as possible fault return messages in the "one-way" operations. This approach would however not work in practice with some of the development platforms, namely "Axis1" and "Metro". Axis2 also encountered a problem processing the *service contract*, with the difference that in this case this shortcoming is considered a bug and has been fixed in the forthcoming version 1.6 (for more info on the problem, please check the following JIRA issue https://issues.apache.org/jira/browse/AXIS2-4408).

4.3.3 List of Operations

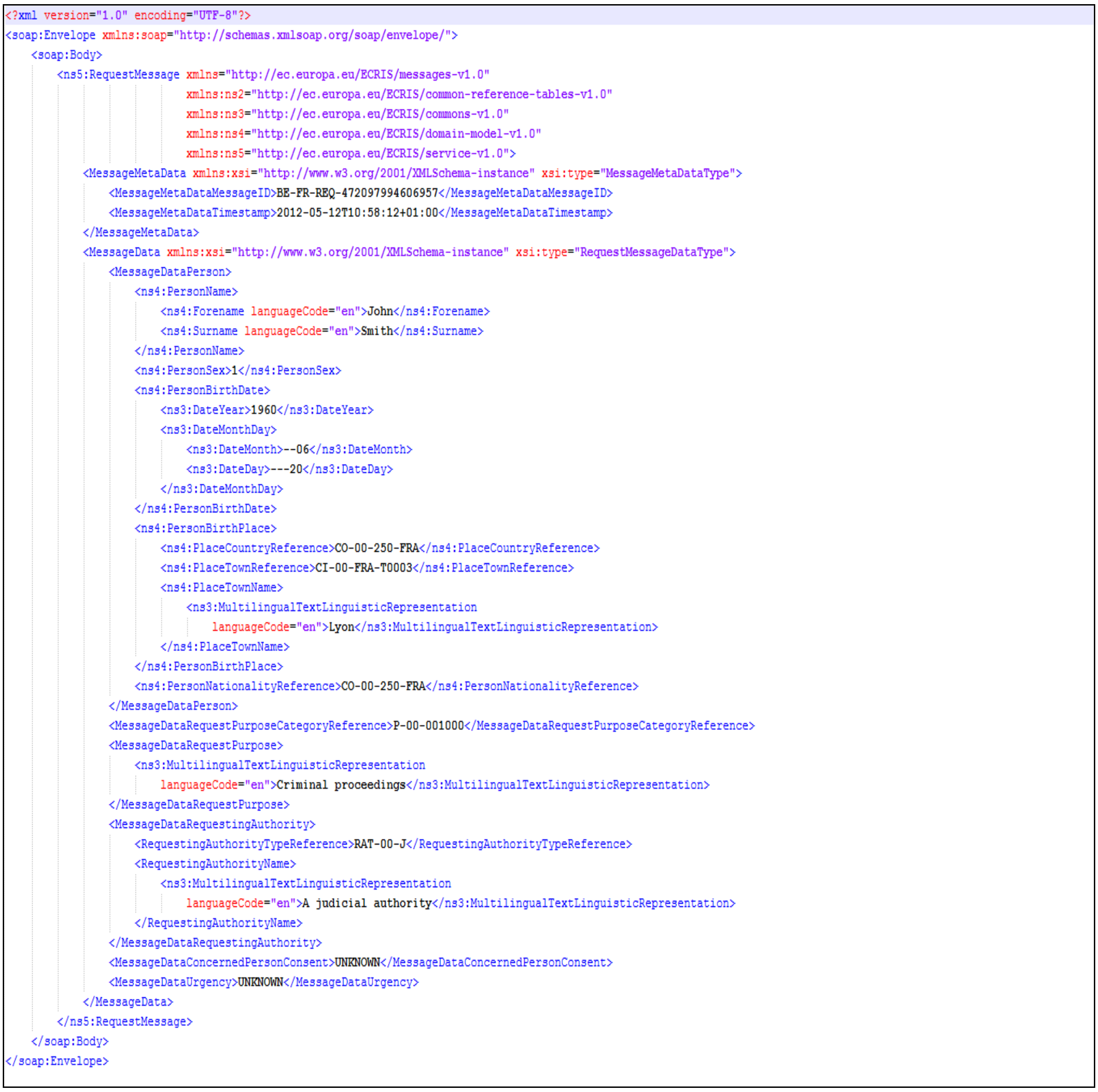
The following sections list the operations defined in the *WSDL* file.

For each operation a description as well as a sample message is presented. Please note that in the sample messages provided, optional elements are frequently left out for improving the readability of this document. In particular the samples used for showing exchanges including binary files are only included partially for the same reasons.

deliverRequest

This operation allows a Member State to transmit a request to another Member State for information and related data to be extracted from the criminal record of a person. It is the entry point that starts an instance of the "Request Criminal Record Information" business process defined in the "Business Analysis" document.

The following figure provides an example of the *SOAP* message:



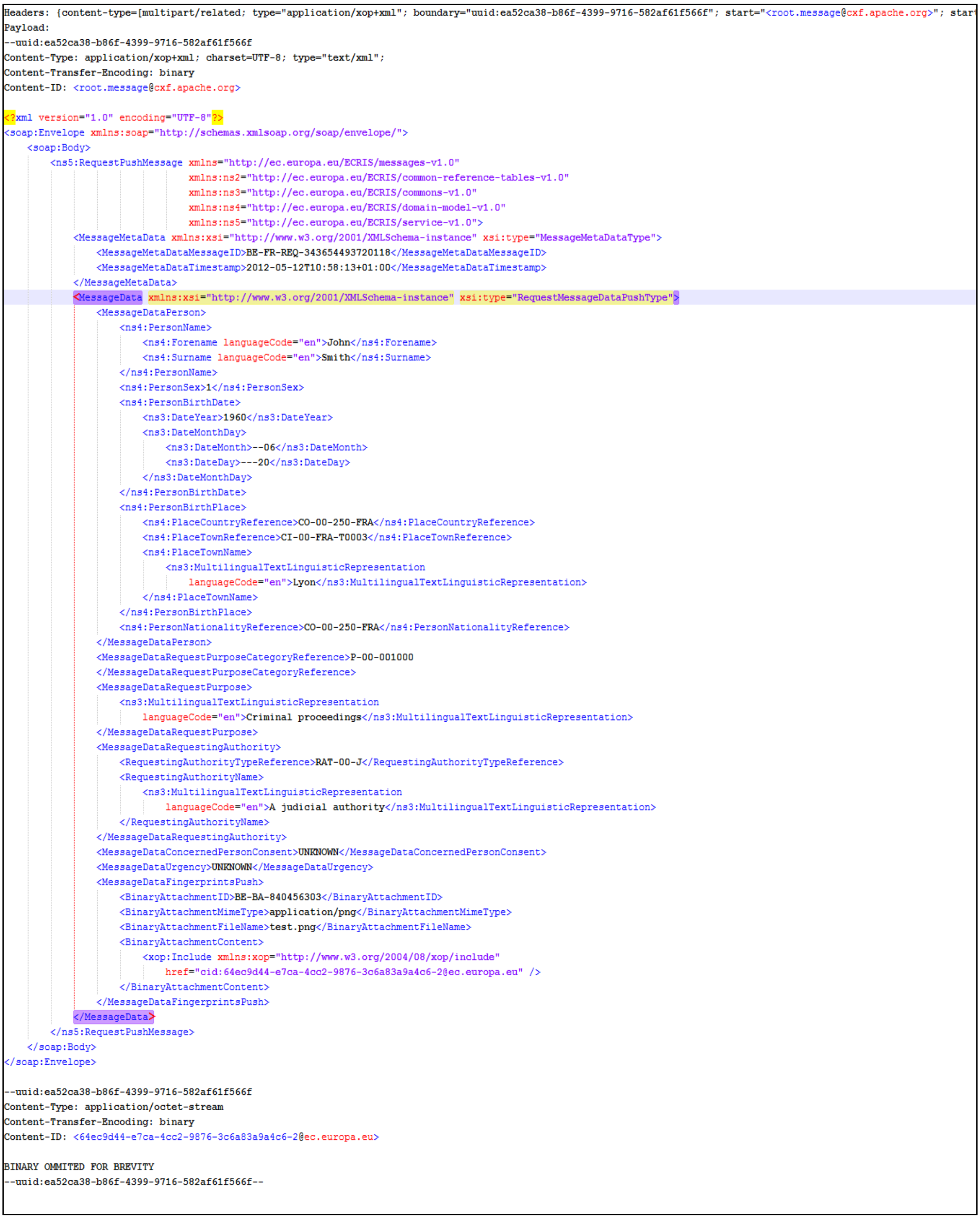
```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                            xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                            xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                            xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-REQ-472097994606957</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:12+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestPurposeCategoryReference>P-00-001000</MessageDataRequestPurposeCategoryReference>
                <MessageDataRequestPurpose>
                    <ns3:MultilingualTextLinguisticRepresentation
                        languageCode="en">Criminal proceedings</ns3:MultilingualTextLinguisticRepresentation>
                </MessageDataRequestPurpose>
                <MessageDataRequestingAuthority>
                    <RequestingAuthorityTypeReference>RAT-00-J</RequestingAuthorityTypeReference>
                    <RequestingAuthorityName>
                        <ns3:MultilingualTextLinguisticRepresentation
                            languageCode="en">A judicial authority</ns3:MultilingualTextLinguisticRepresentation>
                    </RequestingAuthorityName>
                </MessageDataRequestingAuthority>
                <MessageDataConcernedPersonConsent>UNKNOWN</MessageDataConcernedPersonConsent>
                <MessageDataUrgency>UNKNOWN</MessageDataUrgency>
            </MessageData>
        </ns5:RequestMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 14 – "RequestMessage" sample message

deliverRequestPush

This operation is identical to the "deliverRequest" operation but additionally supports sending the binary file as attachment to the "request" message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



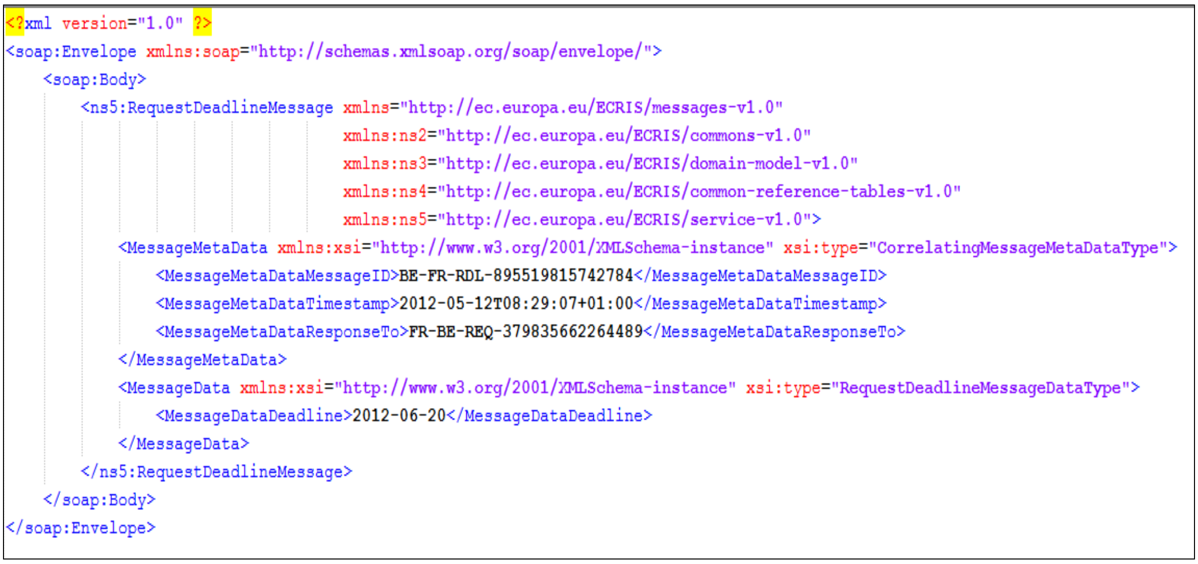
```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f"; start="<root.message@cxf.apache.org>"; star
Payload:
--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestPushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-REQ-343654493720118</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:13+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestMessageDataPushType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestPurposeCategoryReference>P-00-001000
                </MessageDataRequestPurposeCategoryReference>
                <MessageDataRequestPurpose>
                    <ns3:MultilingualTextLinguisticRepresentation
                        languageCode="en">Criminal proceedings</ns3:MultilingualTextLinguisticRepresentation>
                </MessageDataRequestPurpose>
                <MessageDataRequestingAuthority>
                    <RequestingAuthorityTypeReference>RAT-00-J</RequestingAuthorityTypeReference>
                    <RequestingAuthorityName>
                        <ns3:MultilingualTextLinguisticRepresentation
                            languageCode="en">A judicial authority</ns3:MultilingualTextLinguisticRepresentation>
                    </RequestingAuthorityName>
                </MessageDataRequestingAuthority>
                <MessageDataConcernedPersonConsent>UNKNOWN</MessageDataConcernedPersonConsent>
                <MessageDataUrgency>UNKNOWN</MessageDataUrgency>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-840456303</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:RequestPushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu>

BINARY OMMITED FOR BREVITY
--uuid:ea52ca38-b86f-4399-9716-582af61f566f--
```

Figure 15 – "RequestPushMessage" sample message

deliverRequestDeadline

This operation allows a Member State to transmit the legal deadline for the response to the requesting Member State. This operation can be used either directly after a request was received but also after receiving additional identification information through the "deliverRequestAdditionalInfo" operation.

The following figure provides an example of the *SOAP* message:



```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestDeadlineMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RDL-895519815742784</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:07+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-379835662264489</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestDeadlineMessageDataType">
                <MessageDataDeadline>2012-06-20</MessageDataDeadline>
            </MessageData>
        </ns5:RequestDeadlineMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 16 – "RequestDeadlineMessage" sample message

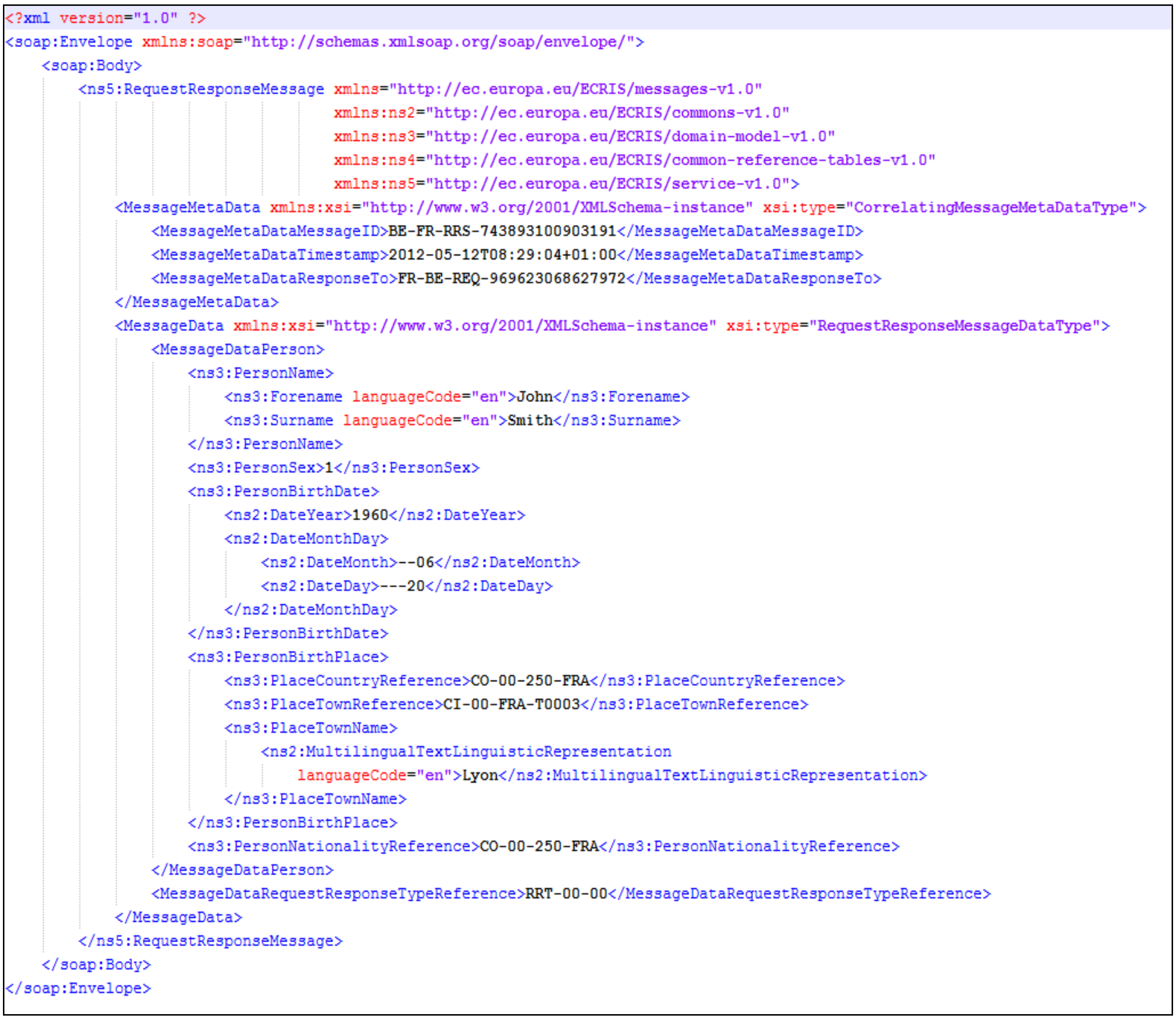
Please note that the legal deadline for the response to the request received is to be calculated implicitly in the local time of the **requested** Member State, and is to be considered **until the end of day, time 23:59:59**, of the transmitted deadline date. For example, in the depiction above, the deadline should be understood as 2012-06-20, until 23:59:59 hours GMT+1.

deliverRequestResponse

This operation allows a Member State to transmit the response to the request, which is the final operation in the "Request Criminal Record Information" business process.

It can contain: (1) information on convictions (if any) in the case that the request could be processed and that a single person matching the identification data has been found, (2) the request denial code if the request has been considered as not valid by the requested Member State or (3) a request problem code if a business issue occurred during the processing of the request.

The following figure provides an example containing no convictions of the *SOAP* message:



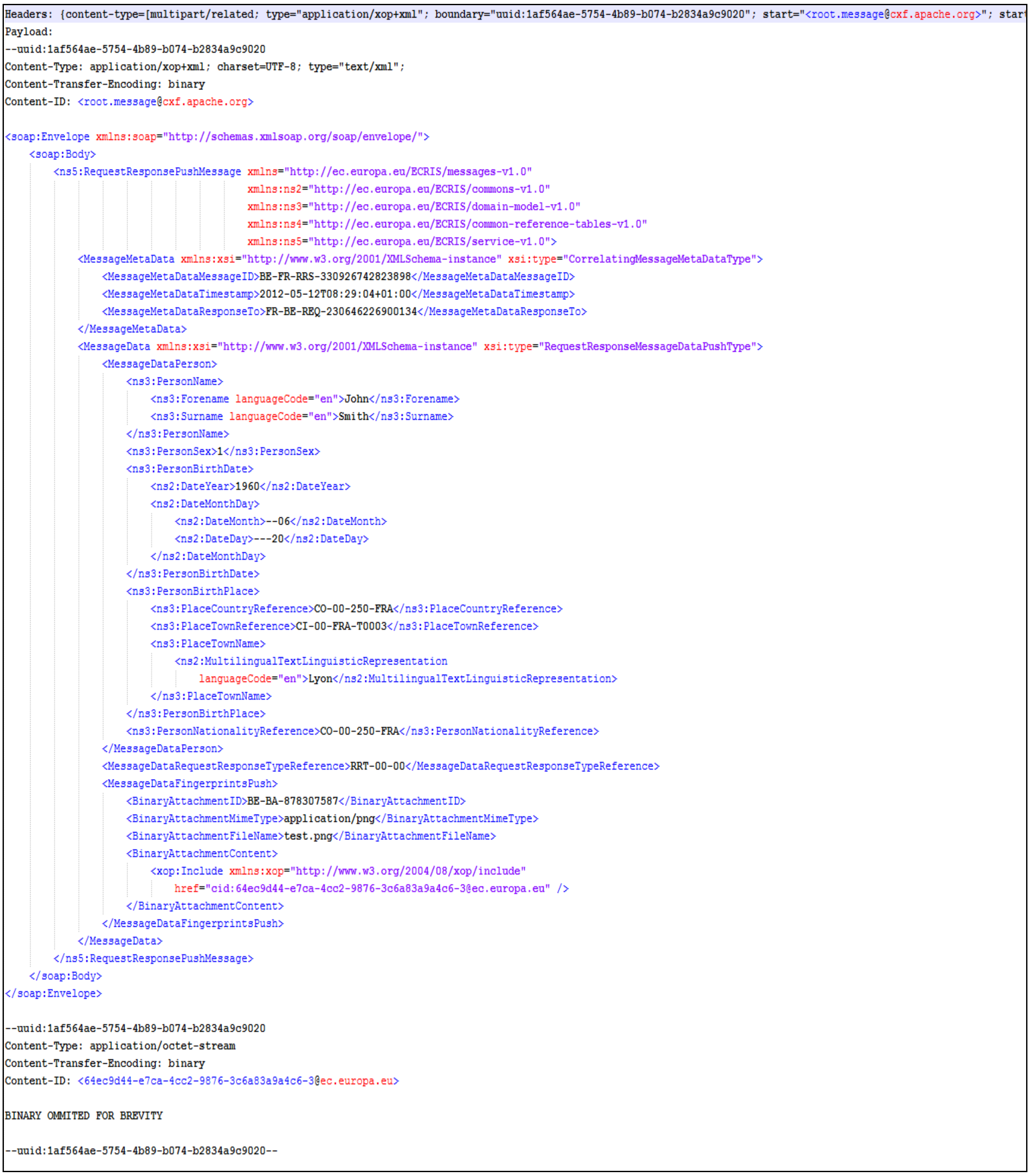
```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RRS-743893100903191</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:04+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-969623068627972</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestResponseMessageDataType">
                <MessageDataPerson>
                    <ns3:PersonName>
                        <ns3:Forename languageCode="en">John</ns3:Forename>
                        <ns3:Surname languageCode="en">Smith</ns3:Surname>
                    </ns3:PersonName>
                    <ns3:PersonSex>1</ns3:PersonSex>
                    <ns3:PersonBirthDate>
                        <ns2:DateYear>1960</ns2:DateYear>
                        <ns2:DateMonthDay>
                            <ns2:DateMonth>--06</ns2:DateMonth>
                            <ns2:DateDay>---20</ns2:DateDay>
                        </ns2:DateMonthDay>
                    </ns3:PersonBirthDate>
                    <ns3:PersonBirthPlace>
                        <ns3:PlaceCountryReference>CO-00-250-FRA</ns3:PlaceCountryReference>
                        <ns3:PlaceTownReference>CI-00-FRA-T0003</ns3:PlaceTownReference>
                        <ns3:PlaceTownName>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:PlaceTownName>
                    </ns3:PersonBirthPlace>
                    <ns3:PersonNationalityReference>CO-00-250-FRA</ns3:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestResponseTypeReference>RRT-00-00</MessageDataRequestResponseTypeReference>
            </MessageData>
        </ns5:RequestResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 17 – "RequestResponseMessage" sample message

deliverRequestResponsePush

This operation is identical to the "deliverRequestResponse" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:1af564ae-5754-4b89-b074-b2834a9c9020"; start="<root.message@cxf.apache.org>"; star
Payload:
--uuid:1af564ae-5754-4b89-b074-b2834a9c9020
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestResponsePushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                        xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                        xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                        xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                        xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RRS-330926742823898</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:04+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-230646226900134</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestResponseMessageDataPushType">
                <MessageDataPerson>
                    <ns3:PersonName>
                        <ns3:Forename languageCode="en">John</ns3:Forename>
                        <ns3:Surname languageCode="en">Smith</ns3:Surname>
                    </ns3:PersonName>
                    <ns3:PersonSex>1</ns3:PersonSex>
                    <ns3:PersonBirthDate>
                        <ns2:DateYear>1960</ns2:DateYear>
                        <ns2:DateMonthDay>
                            <ns2:DateMonth>--06</ns2:DateMonth>
                            <ns2:DateDay>---20</ns2:DateDay>
                        </ns2:DateMonthDay>
                    </ns3:PersonBirthDate>
                    <ns3:PersonBirthPlace>
                        <ns3:PlaceCountryReference>CO-00-250-FRA</ns3:PlaceCountryReference>
                        <ns3:PlaceTownReference>CI-00-FRA-T0003</ns3:PlaceTownReference>
                        <ns3:PlaceTownName>
                            <ns2:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns2:MultilingualTextLinguisticRepresentation>
                        </ns3:PlaceTownName>
                    </ns3:PersonBirthPlace>
                    <ns3:PersonNationalityReference>CO-00-250-FRA</ns3:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataRequestResponseTypeReference>RRT-00-00</MessageDataRequestResponseTypeReference>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-878307587</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-3@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:RequestResponsePushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:1af564ae-5754-4b89-b074-b2834a9c9020
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-3@ec.europa.eu>

BINARY OMMITED FOR BREVITY

--uuid:1af564ae-5754-4b89-b074-b2834a9c9020--
```

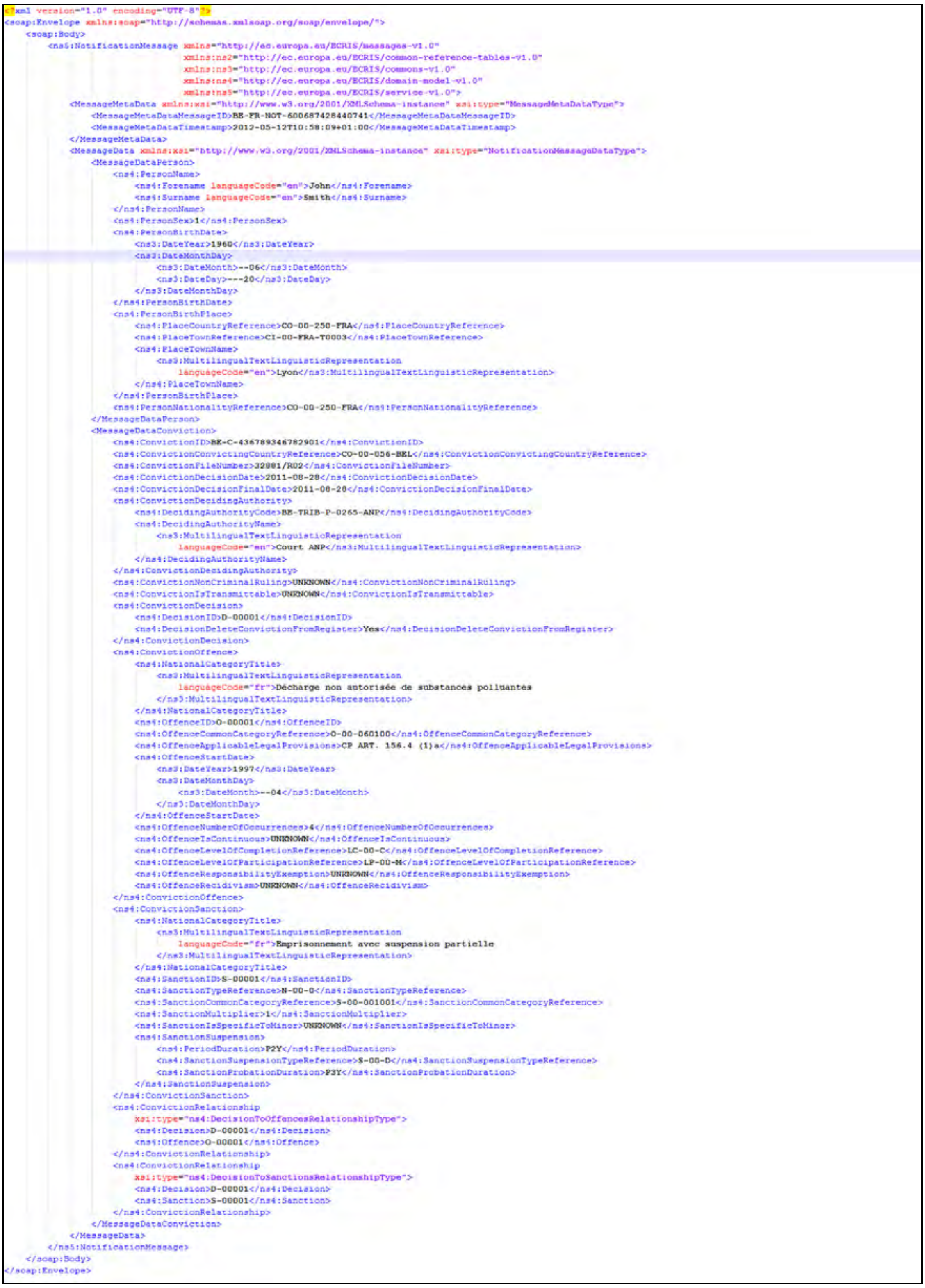
Figure 18 – "RequestResponsePushMessage" sample message

deliverNotification

This operation allows a Member State to transmit a notification to the Member State of the convicted person's nationality, carrying information on the conviction as well as information on subsequent alterations and deletions.

It is the entry point that starts an instance of the "Notify Convictions and Subsequent Changes" business process defined in the "Business Analysis" document.

The following figure provides an example of the *SOAP* message:



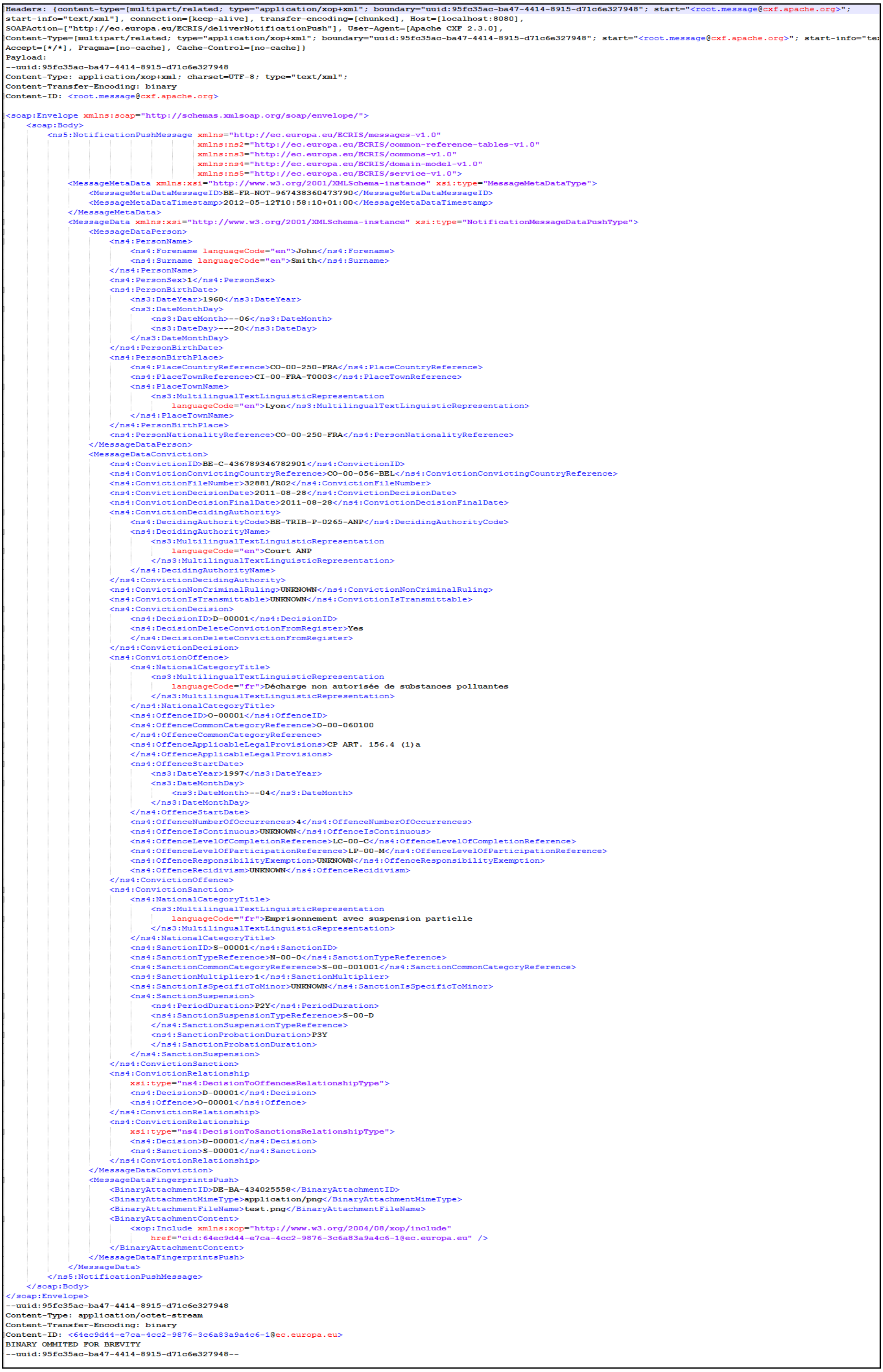
```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                 xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                 xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                 xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                 xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NOT-600687428440741</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:09+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataConviction>
                    <ns4:ConvictionID>BE-C-436789346782901</ns4:ConvictionID>
                    <ns4:ConvictionConvictingCountryReference>CO-00-056-BEL</ns4:ConvictionConvictingCountryReference>
                    <ns4:ConvictionFileNumber>32881/R02</ns4:ConvictionFileNumber>
                    <ns4:ConvictionDecisionDate>2011-08-28</ns4:ConvictionDecisionDate>
                    <ns4:ConvictionDecisionFinalDate>2011-08-28</ns4:ConvictionDecisionFinalDate>
                    <ns4:ConvictionDecidingAuthority>
                        <ns4:DecidingAuthorityCode>BE-TRIB-P-0265-ANP</ns4:DecidingAuthorityCode>
                        <ns4:DecidingAuthorityName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Court ANP</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:DecidingAuthorityName>
                    </ns4:ConvictionDecidingAuthority>
                    <ns4:ConvictionNonCriminalRuling>UNKNOWN</ns4:ConvictionNonCriminalRuling>
                    <ns4:ConvictionIsTransmittable>UNKNOWN</ns4:ConvictionIsTransmittable>
                    <ns4:ConvictionDecision>
                        <ns4:DecisionID>D-00001</ns4:DecisionID>
                        <ns4:DecisionDeleteConvictionFromRegister>Yes</ns4:DecisionDeleteConvictionFromRegister>
                    </ns4:ConvictionDecision>
                    <ns4:ConvictionOffence>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Décharge non autorisée de substances polluantes
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:OffenceID>O-00001</ns4:OffenceID>
                        <ns4:OffenceCommonCategoryReference>O-00-060100</ns4:OffenceCommonCategoryReference>
                        <ns4:OffenceApplicableLegalProvisions>CP ART. 156.4 (1)a</ns4:OffenceApplicableLegalProvisions>
                        <ns4:OffenceStartDate>
                            <ns3:DateYear>1997</ns3:DateYear>
                            <ns3:DateMonthDay>
                                <ns3:DateMonth>--04</ns3:DateMonth>
                            </ns3:DateMonthDay>
                        </ns4:OffenceStartDate>
                        <ns4:OffenceNumberOfOccurrences>4</ns4:OffenceNumberOfOccurrences>
                        <ns4:OffenceIsContinuous>UNKNOWN</ns4:OffenceIsContinuous>
                        <ns4:OffenceLevelOfCompletionReference>LC-00-C</ns4:OffenceLevelOfCompletionReference>
                        <ns4:OffenceLevelOfParticipationReference>LP-00-M</ns4:OffenceLevelOfParticipationReference>
                        <ns4:OffenceResponsibilityExemption>UNKNOWN</ns4:OffenceResponsibilityExemption>
                        <ns4:OffenceRecidivism>UNKNOWN</ns4:OffenceRecidivism>
                    </ns4:ConvictionOffence>
                    <ns4:ConvictionSanction>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Emprisonnement avec suspension partielle
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:SanctionID>S-00001</ns4:SanctionID>
                        <ns4:SanctionTypeReference>N-00-0</ns4:SanctionTypeReference>
                        <ns4:SanctionCommonCategoryReference>S-00-001001</ns4:SanctionCommonCategoryReference>
                        <ns4:SanctionMultiplier>1</ns4:SanctionMultiplier>
                        <ns4:SanctionIsSpecificToMinor>UNKNOWN</ns4:SanctionIsSpecificToMinor>
                        <ns4:SanctionSuspension>
                            <ns4:PeriodDuration>P2Y</ns4:PeriodDuration>
                            <ns4:SanctionSuspensionTypeReference>S-00-D</ns4:SanctionSuspensionTypeReference>
                            <ns4:SanctionProbationDuration>P3Y</ns4:SanctionProbationDuration>
                        </ns4:SanctionSuspension>
                    </ns4:ConvictionSanction>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToOffencesRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Offence>O-00001</ns4:Offence>
                    </ns4:ConvictionRelationship>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToSanctionsRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Sanction>S-00001</ns4:Sanction>
                    </ns4:ConvictionRelationship>
                </MessageDataConviction>
            </MessageData>
        </ns5:NotificationMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 19 – "NotificationMessage" sample message

deliverNotificationPush

This operation is identical to the "deliverNotification" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



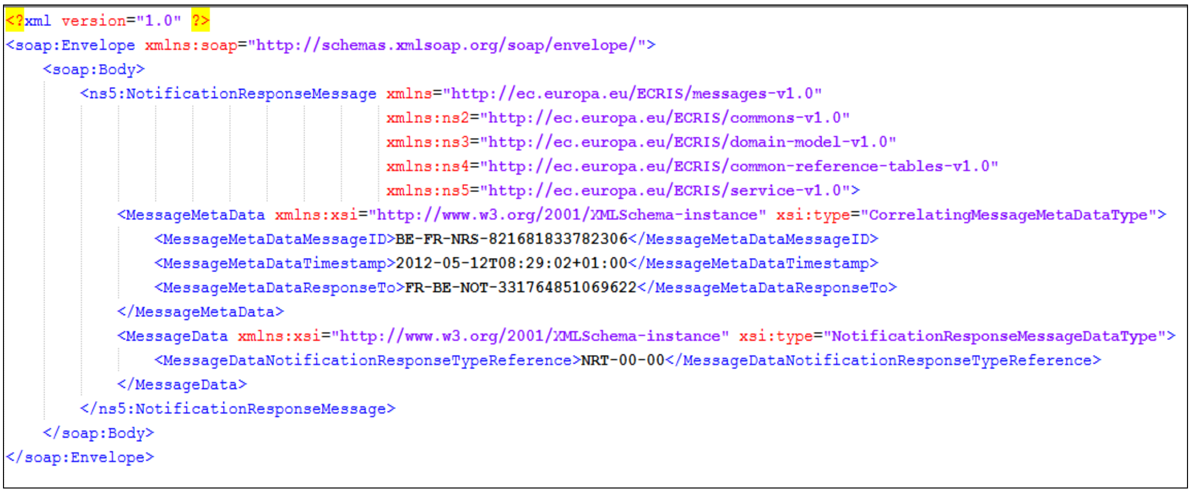
```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:95fc35ac-ba47-4414-8915-d71c6e327948"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], connection=[keep-alive], transfer-encoding=[chunked], Host=[localhost:8080],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverNotificationPush"], User-Agent=[Apache CXF 2.3.0],
Content-Type=[multipart/related; type="application/xop+xml"; boundary="uuid:95fc35ac-ba47-4414-8915-d71c6e327948"; start="<root.message@cxf.apache.org>"; start-info="te
Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationPushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                     xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                     xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                     xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                     xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NOT-967438360473790</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:10+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationMessageDataPushType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                </MessageDataPerson>
                <MessageDataConviction>
                    <ns4:ConvictionID>BE-C-436789346782901</ns4:ConvictionID>
                    <ns4:ConvictionConvictingCountryReference>CO-00-056-BEL</ns4:ConvictionConvictingCountryReference>
                    <ns4:ConvictionFileNumber>32881/R02</ns4:ConvictionFileNumber>
                    <ns4:ConvictionDecisionDate>2011-08-28</ns4:ConvictionDecisionDate>
                    <ns4:ConvictionDecisionFinalDate>2011-08-28</ns4:ConvictionDecisionFinalDate>
                    <ns4:ConvictionDecidingAuthority>
                        <ns4:DecidingAuthorityCode>BE-TRIB-P-0265-ANP</ns4:DecidingAuthorityCode>
                        <ns4:DecidingAuthorityName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Court ANP
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:DecidingAuthorityName>
                    </ns4:ConvictionDecidingAuthority>
                    <ns4:ConvictionNonCriminalRuling>UNKNOWN</ns4:ConvictionNonCriminalRuling>
                    <ns4:ConvictionIsTransmittable>UNKNOWN</ns4:ConvictionIsTransmittable>
                    <ns4:ConvictionDecision>
                        <ns4:DecisionID>D-00001</ns4:DecisionID>
                        <ns4:DecisionDeleteConvictionFromRegister>Yes
                        </ns4:DecisionDeleteConvictionFromRegister>
                    </ns4:ConvictionDecision>
                    <ns4:ConvictionOffence>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Décharge non autorisée de substances polluantes
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:OffenceID>O-00001</ns4:OffenceID>
                        <ns4:OffenceCommonCategoryReference>O-00-060100
                        </ns4:OffenceCommonCategoryReference>
                        <ns4:OffenceApplicableLegalProvisions>CP ART. 156.4 (1)a
                        </ns4:OffenceApplicableLegalProvisions>
                        <ns4:OffenceStartDate>
                            <ns3:DateYear>1997</ns3:DateYear>
                            <ns3:DateMonthDay>
                                <ns3:DateMonth>--04</ns3:DateMonth>
                            </ns3:DateMonthDay>
                        </ns4:OffenceStartDate>
                        <ns4:OffenceNumberOfOccurrences>4</ns4:OffenceNumberOfOccurrences>
                        <ns4:OffenceIsContinuous>UNKNOWN</ns4:OffenceIsContinuous>
                        <ns4:OffenceLevelOfCompletionReference>LC-00-C</ns4:OffenceLevelOfCompletionReference>
                        <ns4:OffenceLevelOfParticipationReference>LP-00-M</ns4:OffenceLevelOfParticipationReference>
                        <ns4:OffenceResponsibilityExemption>UNKNOWN</ns4:OffenceResponsibilityExemption>
                        <ns4:OffenceRecidivism>UNKNOWN</ns4:OffenceRecidivism>
                    </ns4:ConvictionOffence>
                    <ns4:ConvictionSanction>
                        <ns4:NationalCategoryTitle>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="fr">Emprisonnement avec suspension partielle
                            </ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:NationalCategoryTitle>
                        <ns4:SanctionID>S-00001</ns4:SanctionID>
                        <ns4:SanctionTypeReference>N-00-0</ns4:SanctionTypeReference>
                        <ns4:SanctionCommonCategoryReference>S-00-001001</ns4:SanctionCommonCategoryReference>
                        <ns4:SanctionMultiplier>1</ns4:SanctionMultiplier>
                        <ns4:SanctionIsSpecificToMinor>UNKNOWN</ns4:SanctionIsSpecificToMinor>
                        <ns4:SanctionSuspension>
                            <ns4:PeriodDuration>P2Y</ns4:PeriodDuration>
                            <ns4:SanctionSuspensionTypeReference>S-00-D
                            </ns4:SanctionSuspensionTypeReference>
                            <ns4:SanctionProbationDuration>P3Y
                            </ns4:SanctionProbationDuration>
                        </ns4:SanctionSuspension>
                    </ns4:ConvictionSanction>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToOffencesRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Offence>O-00001</ns4:Offence>
                    </ns4:ConvictionRelationship>
                    <ns4:ConvictionRelationship
                        xsi:type="ns4:DecisionToSanctionsRelationshipType">
                        <ns4:Decision>D-00001</ns4:Decision>
                        <ns4:Sanction>S-00001</ns4:Sanction>
                    </ns4:ConvictionRelationship>
                </MessageDataConviction>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>DE-BA-434025558</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-1@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:NotificationPushMessage>
    </soap:Body>
</soap:Envelope>
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-1@ec.europa.eu>
BINARY OMMITED FOR BREVITY
--uuid:95fc35ac-ba47-4414-8915-d71c6e327948--
```

Figure 20 – "NotificationPushMessage" sample message

deliverNotificationResponse

This operation allows a Member State to inform the convicting Member State of the outcome of the processing of the notification. It is the final operation in the "Notify Convictions and Subsequent Changes" business process. The message can contain: (1) a success message and optionally the identification information under which the notification has been processed or (2) a problem code if a business issue occurred during the processing of the notification.

The following figure provides an example of the *SOAP* message:



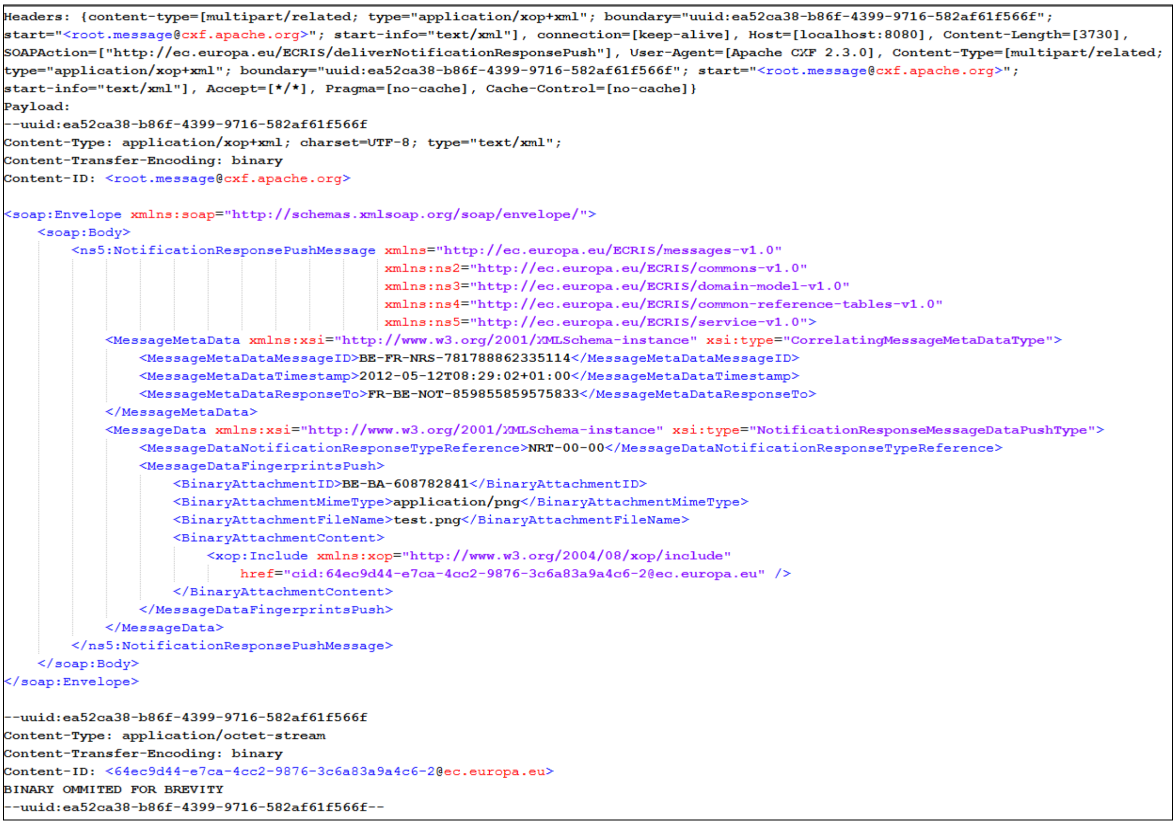
```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                         xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                         xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                         xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                         xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NRS-821681833782306</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:02+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-NOT-331764851069622</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationResponseMessageDataType">
                <MessageDataNotificationResponseTypeReference>NRT-00-00</MessageDataNotificationResponseTypeReference>
            </MessageData>
        </ns5:NotificationResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 21 – "NotificationResponseMessage" sample message

deliverNotificationResponsePush

This operation is identical to the “deliverNotificationResponse” operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the “Push” approach that has been in the “Technical Architecture” document).

The following figure provides an example of the *SOAP* message:



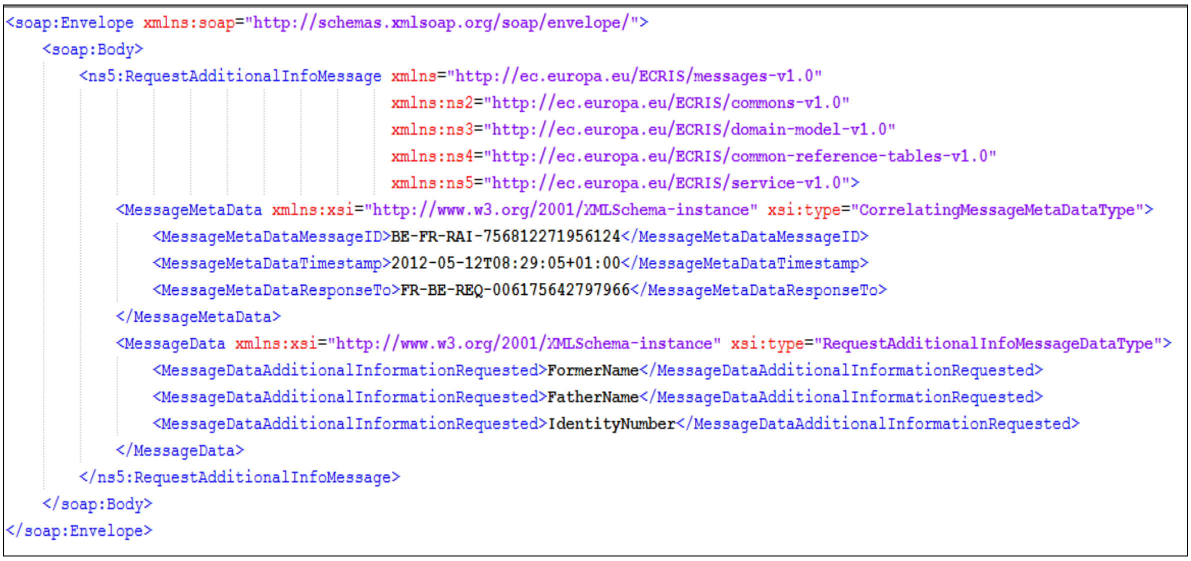
```
Headers: {content-type=[multipart/related; type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f";
start="<root.message@cxf.apache.org>"; start-info="text/xml"], connection=[keep-alive], Host=[localhost:8080], Content-Length=[3730],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverNotificationResponsePush"], User-Agent=[Apache CXF 2.3.0], Content-Type=[multipart/related;
type="application/xop+xml"; boundary="uuid:ea52ca38-b86f-4399-9716-582af61f566f"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-cache]}
Payload:
--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:NotificationResponsePushMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                             xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                             xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                             xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                             xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-NRS-781788862335114</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:02+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-NOT-859855859575833</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="NotificationResponseMessageDataPushType">
                <MessageDataNotificationResponseTypeReference>NRT-00-00</MessageDataNotificationResponseTypeReference>
                <MessageDataFingerprintsPush>
                    <BinaryAttachmentID>BE-BA-608782841</BinaryAttachmentID>
                    <BinaryAttachmentMimeType>application/png</BinaryAttachmentMimeType>
                    <BinaryAttachmentFileName>test.png</BinaryAttachmentFileName>
                    <BinaryAttachmentContent>
                        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
                            href="cid:64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu" />
                    </BinaryAttachmentContent>
                </MessageDataFingerprintsPush>
            </MessageData>
        </ns5:NotificationResponsePushMessage>
    </soap:Body>
</soap:Envelope>

--uuid:ea52ca38-b86f-4399-9716-582af61f566f
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-2@ec.europa.eu>
BINARY OMMITED FOR BREVITY
--uuid:ea52ca38-b86f-4399-9716-582af61f566f--
```

Figure 22 – “NotificationResponsePushMessage” sample message

deliverRequestAdditionalInfo

This operation is used by the Member State performing the search of a person for asking additional identification information to the requesting/convicting Member State.

The following figure provides an example of the *SOAP* message:



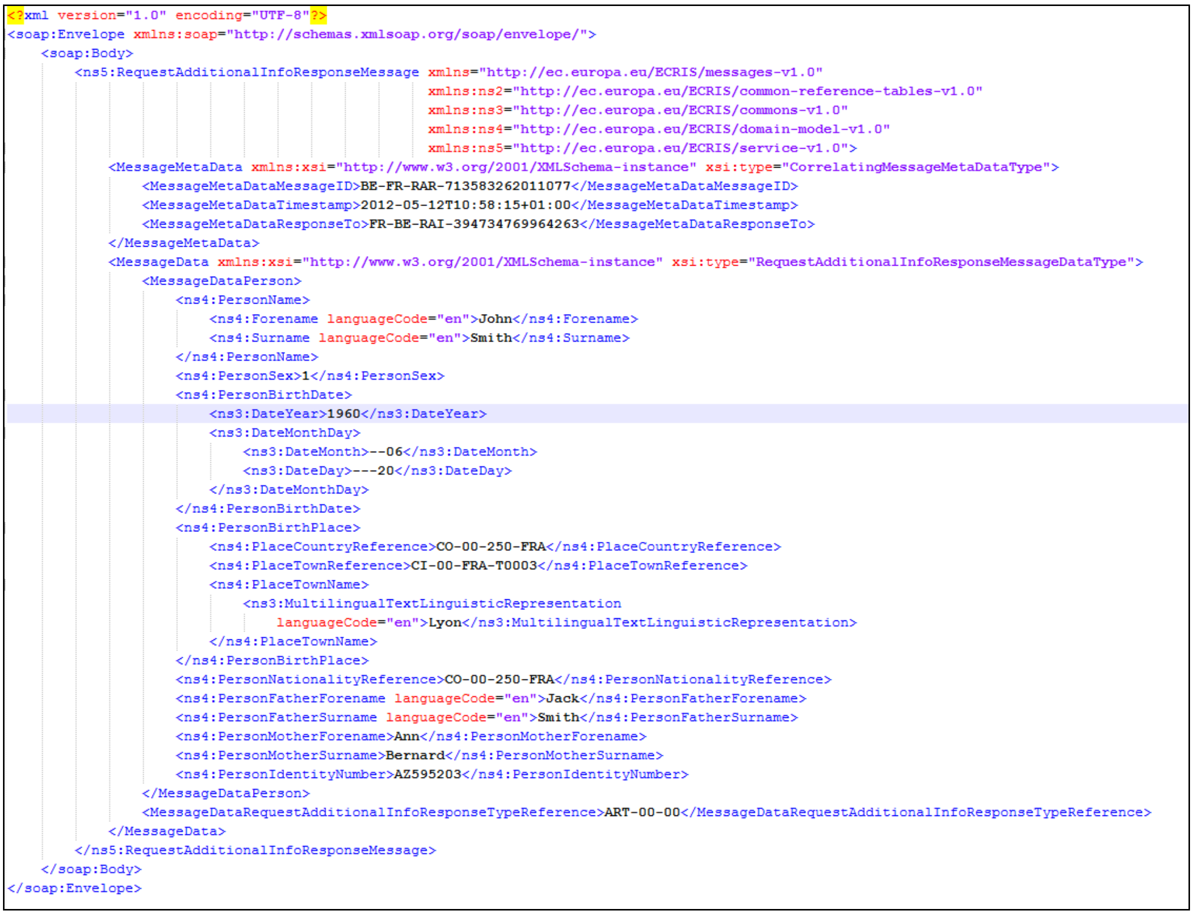
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                          xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                          xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                          xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                          xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAI-756812271956124</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:05+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-006175642797966</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoMessageDataType">
                <MessageDataAdditionalInformationRequested>FormerName</MessageDataAdditionalInformationRequested>
                <MessageDataAdditionalInformationRequested>FatherName</MessageDataAdditionalInformationRequested>
                <MessageDataAdditionalInformationRequested>IdentityNumber</MessageDataAdditionalInformationRequested>
            </MessageData>
        </ns5:RequestAdditionalInfoMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 23 – "RequestAdditionalInfoMessage" sample message

deliverRequestAdditionalInfoResponse

This operation allows a Member State to respond to the "request additional info" message, either with updated personal identification data or with an indication that no additional information is available.

The following figure provides an example of the *SOAP* message:



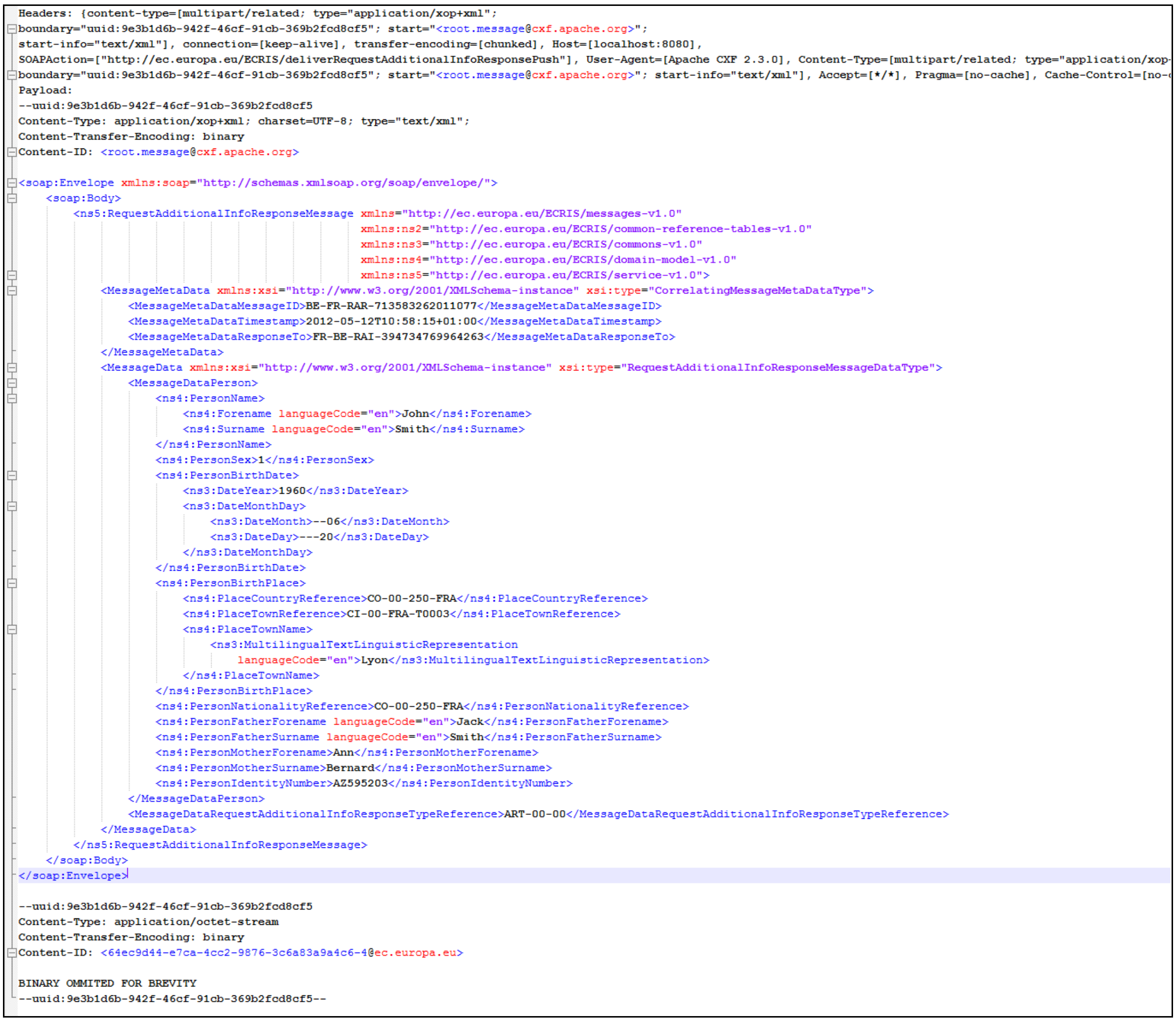
```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                                  xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                                  xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                                  xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                                  xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAR-713583262011077</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:15+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-RAI-394734769964263</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoResponseMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                    <ns4:PersonFatherForename languageCode="en">Jack</ns4:PersonFatherForename>
                    <ns4:PersonFatherSurname languageCode="en">Smith</ns4:PersonFatherSurname>
                    <ns4:PersonMotherForename>Ann</ns4:PersonMotherForename>
                    <ns4:PersonMotherSurname>Bernard</ns4:PersonMotherSurname>
                    <ns4:PersonIdentityNumber>AZ595203</ns4:PersonIdentityNumber>
                </MessageDataPerson>
                <MessageDataRequestAdditionalInfoResponseTypeReference>ART-00-00</MessageDataRequestAdditionalInfoResponseTypeReference>
            </MessageData>
        </ns5:RequestAdditionalInfoResponseMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 24 – “RequestAdditionalInfoResponseMessage” sample message

deliverRequestAdditionalInfoResponsePush

This operation is identical to the "deliverRequestAdditionalInfoResponse" operation but additionally supports sending the binary file as attachment to the message (i.e. implementation of the "Push" approach that has been in the "Technical Architecture" document).

The following figure provides an example of the *SOAP* message:



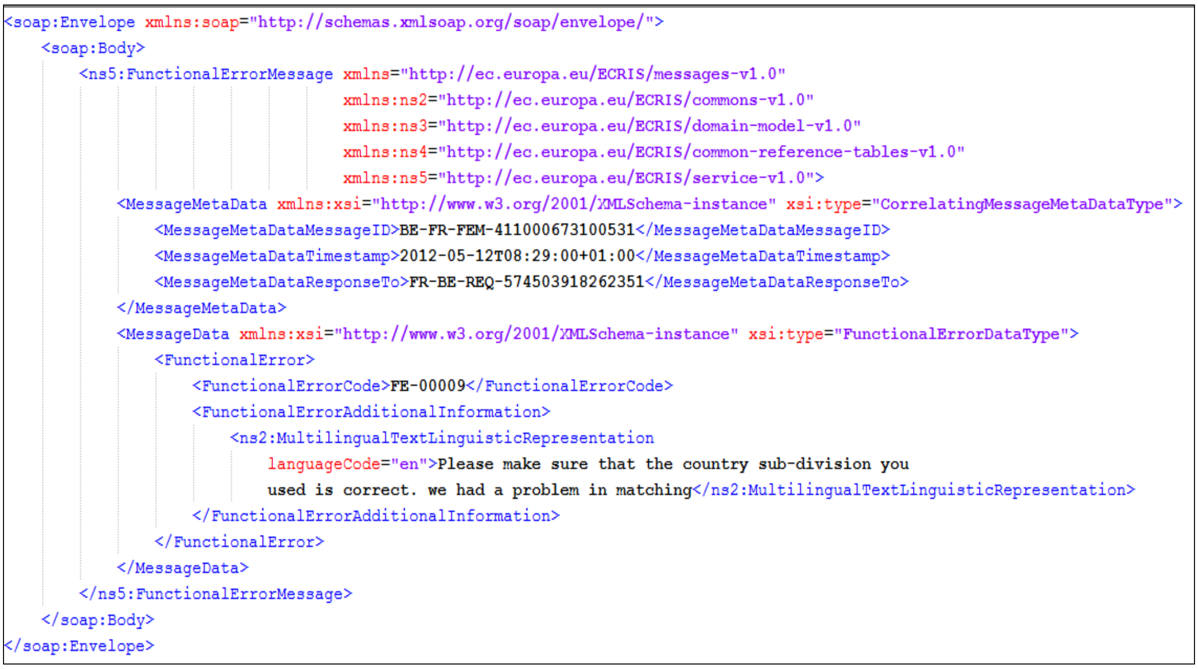
```
Headers: {content-type=[multipart/related; type="application/xop+xml";
boundary="uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5"; start="<root.message@cxf.apache.org>";
start-info="text/xml"], connection=[keep-alive], transfer-encoding=[chunked], Host=[localhost:8080],
SOAPAction=["http://ec.europa.eu/ECRIS/deliverRequestAdditionalInfoResponsePush"], User-Agent=[Apache CXF 2.3.0], Content-Type=[multipart/related; type="application/xop-
boundary="uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5"; start="<root.message@cxf.apache.org>"; start-info="text/xml"], Accept=[*/*], Pragma=[no-cache], Cache-Control=[no-
Payload:
--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5
Content-Type: application/xop+xml; charset=UTF-8; type="text/xml";
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:RequestAdditionalInfoResponseMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                                  xmlns:ns2="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                                  xmlns:ns3="http://ec.europa.eu/ECRIS/commons-v1.0"
                                                  xmlns:ns4="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                                  xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-RAR-713583262011077</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T10:58:15+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-RAI-394734769964263</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="RequestAdditionalInfoResponseMessageDataType">
                <MessageDataPerson>
                    <ns4:PersonName>
                        <ns4:Forename languageCode="en">John</ns4:Forename>
                        <ns4:Surname languageCode="en">Smith</ns4:Surname>
                    </ns4:PersonName>
                    <ns4:PersonSex>1</ns4:PersonSex>
                    <ns4:PersonBirthDate>
                        <ns3:DateYear>1960</ns3:DateYear>
                        <ns3:DateMonthDay>
                            <ns3:DateMonth>--06</ns3:DateMonth>
                            <ns3:DateDay>---20</ns3:DateDay>
                        </ns3:DateMonthDay>
                    </ns4:PersonBirthDate>
                    <ns4:PersonBirthPlace>
                        <ns4:PlaceCountryReference>CO-00-250-FRA</ns4:PlaceCountryReference>
                        <ns4:PlaceTownReference>CI-00-FRA-T0003</ns4:PlaceTownReference>
                        <ns4:PlaceTownName>
                            <ns3:MultilingualTextLinguisticRepresentation
                                languageCode="en">Lyon</ns3:MultilingualTextLinguisticRepresentation>
                        </ns4:PlaceTownName>
                    </ns4:PersonBirthPlace>
                    <ns4:PersonNationalityReference>CO-00-250-FRA</ns4:PersonNationalityReference>
                    <ns4:PersonFatherForename languageCode="en">Jack</ns4:PersonFatherForename>
                    <ns4:PersonFatherSurname languageCode="en">Smith</ns4:PersonFatherSurname>
                    <ns4:PersonMotherForename>Ann</ns4:PersonMotherForename>
                    <ns4:PersonMotherSurname>Bernard</ns4:PersonMotherSurname>
                    <ns4:PersonIdentityNumber>AZ595203</ns4:PersonIdentityNumber>
                </MessageDataPerson>
                <MessageDataRequestAdditionalInfoResponseTypeReference>ART-00-00</MessageDataRequestAdditionalInfoResponseTypeReference>
            </MessageData>
        </ns5:RequestAdditionalInfoResponseMessage>
    </soap:Body>
</soap:Envelope>

--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <64ec9d44-e7ca-4cc2-9876-3c6a83a9a4c6-4@ec.europa.eu>

BINARY OMMITED FOR BREVITY
--uuid:9e3b1d6b-942f-46cf-91cb-369b2fcd8cf5--
```

Figure 25 – "RequestAdditionalInfoResponsePushMessage" sample message

deliverFunctionalError

This operation allows a Member State to provide information to a sending Member State on a functional error that occurred during the processing of a previous ECRIS message. It can be used at any time once an instance of a business process started and before it finishes.

The following figure provides an example of the *SOAP* message:



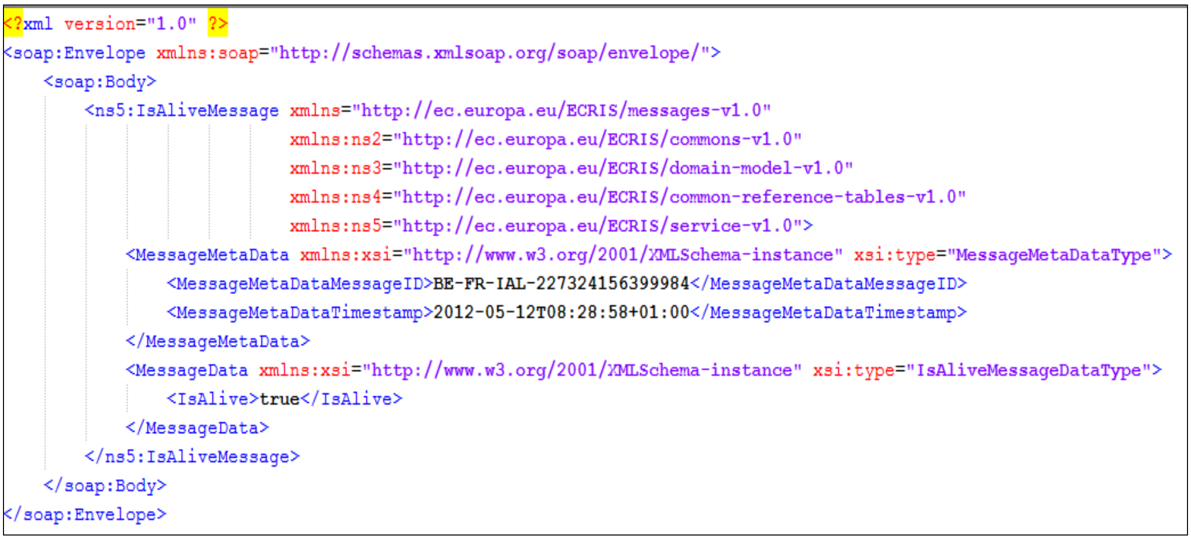
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:FunctionalErrorMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                                    xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                                    xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                                    xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                                    xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CorrelatingMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-FEM-411000673100531</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:29:00+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>FR-BE-REQ-574503918262351</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="FunctionalErrorDataType">
                <FunctionalError>
                    <FunctionalErrorCode>FE-00009</FunctionalErrorCode>
                    <FunctionalErrorAdditionalInformation>
                        <ns2:MultilingualTextLinguisticRepresentation
                            languageCode="en">Please make sure that the country sub-division you
                            used is correct. we had a problem in matching</ns2:MultilingualTextLinguisticRepresentation>
                    </FunctionalErrorAdditionalInformation>
                </FunctionalError>
            </MessageData>
        </ns5:FunctionalErrorMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 26 – "FunctionalErrorMessage" sample message

isAlive

The "isAlive" operation allows checking whether another ECRIS *web service* endpoint is properly responding to calls.

The following figure provides an example of the *SOAP* message:



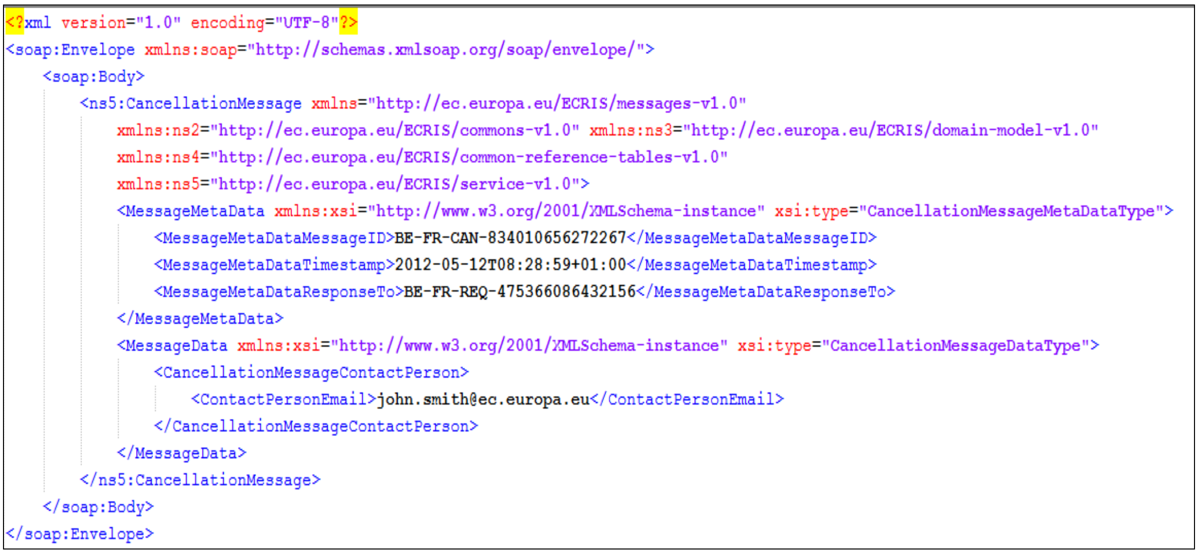
```
<?xml version="1.0" ?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:IsAliveMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
                            xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0"
                            xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
                            xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
                            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="MessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-IAL-227324156399984</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:28:58+01:00</MessageMetaDataTimestamp>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="IsAliveMessageDataType">
                <IsAlive>true</IsAlive>
            </MessageData>
        </ns5:IsAliveMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 27 – "isAlive" sample message

deliverCancellation

The "deliverCancellation" is the operation that allows cancelling a previously sent message of type request, request response, notification or notification response.

The following figure provides an example of the *SOAP* message:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <ns5:CancellationMessage xmlns="http://ec.europa.eu/ECRIS/messages-v1.0"
            xmlns:ns2="http://ec.europa.eu/ECRIS/commons-v1.0" xmlns:ns3="http://ec.europa.eu/ECRIS/domain-model-v1.0"
            xmlns:ns4="http://ec.europa.eu/ECRIS/common-reference-tables-v1.0"
            xmlns:ns5="http://ec.europa.eu/ECRIS/service-v1.0">
            <MessageMetaData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CancellationMessageMetaDataType">
                <MessageMetaDataMessageID>BE-FR-CAN-834010656272267</MessageMetaDataMessageID>
                <MessageMetaDataTimestamp>2012-05-12T08:28:59+01:00</MessageMetaDataTimestamp>
                <MessageMetaDataResponseTo>BE-FR-REQ-475366086432156</MessageMetaDataResponseTo>
            </MessageMetaData>
            <MessageData xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="CancellationMessageDataType">
                <CancellationMessageContactPerson>
                    <ContactPersonEmail>john.smith@ec.europa.eu</ContactPersonEmail>
                </CancellationMessageContactPerson>
            </MessageData>
        </ns5:CancellationMessage>
    </soap:Body>
</soap:Envelope>
```

Figure 28 – "deliverCancellation" sample message

4.4 Handling of Errors

Apart from the business errors which are dealt with by appropriate messages, two types of error responses are supported by the ECRIS technical specifications:

- Synchronous *SOAP* fault messages in response to technical errors.
- Asynchronous functional error messages in response to functional errors.

4.4.1 Technical Errors

As explained previously, although the *web service* operations are defined as "one-way" operations and do not explicitly declare faults as possible returns, the ECRIS software implementation **must throw synchronously a *SOAP* fault** when encountering a low-level technical error (i.e. connectivity errors, XML message does not validate against the XSD, etc.).

Please note that the development platforms that have been tested and that are described later in this document are capable of properly throwing *SOAP* faults when an exception occurs, even if the *SOAP* fault is not explicitly declared in the *WSDL* definition. In addition to this, and because the structural validation of the XML message is performed synchronously when receiving the XML stream, the *web service* implementation automatically throws an adequate technical error if the XML does not respect the *XSD* definitions. In such a case, the error message normally explicitly and clearly indicates which XML element is problematic and why. In such a case, the stack-trace indicates clearly which XML element is problematic and why. It must also be noted that the structure of the standard SOAP faults allow providing a custom error code, a stack-trace and additional custom information that the implementer wishes to log.

4.4.2 Functional Errors

As defined in the "Technical Architecture" document, once the technical transmission of a message has been performed successfully, a **functional validation** of the content of the message can take place **in a second step** and may result in a functional error to be returned

asynchronously to the sender of the message if a problem is encountered. The detailed list of cases that are to provoke such functional errors is provided later in this document. For each such error condition, an appropriate functional error code is defined in the common reference table "FunctionalErrors".

When a functional error occurs, the message to which it relates is to be rejected. Such functional errors are defined for trapping cases of severe inconsistency of the information being transmitted, so as to capture sufficiently early bugs in the software or obvious end-user mistakes. It is expected that when such a functional error occurs, the situation can easily be corrected by the sender who can simply resend the corrected faulty message.

Please note that the functional error message can contain several error codes. Indeed, it is not the aim of sending multiple functional error messages if several problems are detected in one message. If this occurs, only one functional message is to be sent with multiple error codes.

In terms of the **processing** of the functional errors, the following rules are defined:

- A functional error can be sent at any point during a transaction as a response to any message received, but **not after the end** of the transaction or before the start of the transaction. In essence, once a message is received that closes a transaction (i.e. a request response or notification response message), the receiver cannot reply with a functional error message.
- The previous rule implies that if there is a problem in a request response or notification response, it needs to be handled manually by the central authorities outside of ECRIS by human operators. To that end, the ECRIS software on the receiver's end should raise an alert to the staff of the central authority so that the issue can be solved. Please note though that the message cancellation described later in this document facilitates the resolution of such a situation. Indeed, the operator on the receiver's end can inform by any communication means available the sender of the problem that has been encountered. The sender then simply cancels the problematic request/notification response and sends a corrected message using the ECRIS software.
- A functional error message ends the transaction in the following cases:
  - when issued as a response to a notification message
  - when issued as a response to a request message, independently of whether the "request deadline" message has been sent before or not; concretely, this means that both sequences "request → functional error" and "request → request deadline → functional error" lead to the end of the transaction
- A functional error message cannot be answered with another functional error (this avoids endless loops).
- Given the previous rule, a message following a functional error must be correlated with the last message received in the transaction and that was not a functional error. For example, Member State "A" sends a request to Member State "B". Member State "B" sends back the request deadline. Then "B" delivers a request for additional information to "A". Let's assume that the response of "A" to the request for additional information of "B" is problematic and triggers a functional error. "A" receives the functional error and re-issues a new corrected response to the request for additional information. This response message must be correlated with the request for additional information message issued by "B".
- Considering the rules above, the kinematics of ECRIS could potentially generate an infinite loop if the sending of additional identification information systematically raises a functional error. Concretely, if the ECRIS processing is fully automated, the following could occur: request → request deadline → request for additional ID info → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → functional error → provide additional ID info → …

  In order to avoid this situation, in the case that the "additional ID info" message raises the same functional error 3 times consecutively within one transaction, the 3rd time the receiving ECRIS software **must drop** the "additional ID info" message without replying to it and consider that no additional information is available. This allows the receiving

Member State to exit the loop and to continue the request or notification process so as to ~~read~~ reach the proper end of the transaction.

4.4.3 Message Cancellation

As described in the list of *web service* operations, the ECRIS detailed technical specifications include a specific cancellation message named "deliverCancellation" that allows annulling a previously sent message so as to be able to correct mistakes.

Please note first that "cancellation" is defined in this context as the **annulment** of a previously transmitted ECRIS **message** because it was sent wrongfully or because it contains erroneous information (i.e. as a result of a bug in the system or of a wrong manipulation by an end-user of the system). It must not be confused with the judicial means for cancelling convictions. This is handled through the rehabilitation and judicial annulment mechanisms. The domain model already foresees the necessary elements for informing the Member State of nationality that a conviction has been rehabilitated or judicially annulled and that it is to be removed from the criminal records register. The "cancellation" is to be understood here on the technical level and as a functionality allowing correcting operational or technical mistakes.

The following rules apply for the usage of this "deliverCancellation" operation:

- **Only** the following types of messages can be cancelled:
  - Request
  - Request response
  - Notification
  - Notification response
- Trying to cancel a message which does not belong to the list above results in a functional error (such a functional error is actually the only direct response that can be provided to a cancellation message).
- The cancellation message itself can **never** be cancelled by a subsequent cancellation message.
- **Only** the sender of a message is allowed to cancel that message.
- Cancelling a request response or a notification response message in a given transaction does not stop the transaction itself while cancelling a request or a notification message stops the entire transaction.

  Thus, in the situation where a request or notification message is cancelled by the requesting or notifying Member State, the transaction is considered as closed and therefore no other messages are expected to be exchanged (e.g. no notification response, request response, request deadline or request for additional information). If such messages are still received as a response to a message that has been cancelled, they are to be handled as unexpected messages, as described later in chapter §4.6.1.
- In the case where a notification response or a request response message is being cancelled, the transaction is reopened. Please note that the **initial deadline** for the transaction **is still valid and is not reset** by cancelling the notification/request response. In such a case, a new message of the same type as the cancelled one, carrying the corrected set of information, is expected to be sent so as to close again the transaction within its deadline. The possibility to cancel such messages is however further limited in time:
  - A request response can be cancelled within one week (i.e. 7 days) after it was sent, provided that the initial transaction deadline has not elapsed.
  - A notification response can be cancelled within one month (i.e. 30 days) after it was sent, provided that the initial transaction deadline has not elapsed.

  For example, if a "request response" message is cancelled, the expected behaviour is to resend a new "request response" message for correcting the cancelled one. When re-sending a corrected message it must however be noted that the technical message ID of the new message must be changed so as to remain unique across all ECRIS messages. This message cannot reuse the unique technical identifier of the cancelled message.

Please note also that if the notification/request response message is cancelled at a moment in time that is very close to the end of the deadline of the notification/request transaction, there may be no sufficient time for actually resending the corrected notification/request response message. In such a borderline case, the central authority having cancelled the problematic message must take contact with the corresponding Member State's central authority in order to solve the issue manually.

- Given the previous rules, a corrected request response or notification response message that is sent after having been cancelled must correlate with the last message received in the transaction and that was not a functional error or the cancellation message itself. For example, Member State "A" sends a request to Member State "B". Member State "B" sends back the request deadline and then the request response. In the central authority of "B", an operator realises that the request response has a severe mistake. "B" sends a cancellation message to "A" for annulling the request response that was just sent. When "B" sends the corrected request response message to "A", it must correlate with the request message issued by "A" since this is the last business message received by "B" in this transaction.
- The previous rules also imply that if a problem is detected for a request response or notification response message after the cancellation time limits defined above, the central authority having sent the problematic message can no longer cancel it via ECRIS but must take contact with the corresponding Member State's central authority in order to solve the issue manually.

Please note that the cancellation mechanism is meant to be used for correcting significant mistakes, such as providing the wrong identity of a person or the wrong list of convictions. Minor corrections to notification messages can be easily transmitted by simply resending the corrected notification message and indicating that this new conviction information updates conviction information previously sent.

Since a cancellation message is **not to be used** for indicating a judicial removal of a conviction but only to correct erroneous situations that occur by mistake, the receiver of such a cancellation message knows for sure that the message that the sender wants to cancel must be dropped and that the information contained in that message must simply be ignored or deleted.

4.5 Handling of Binary Attachments

The ECRIS detailed technical specifications include the support for exchanging fingerprints in the form of binary attachments to the XML messages. These binary attachments are optional and may only carry NIST files, as defined in the "Technical Architecture" and "Business Analysis" documents.

More specifically, the "Technical Architecture" document states:

"As a result, the optional exchange of fingerprints in ECRIS is to be performed by joining NIST files as binary attachments to the *XML* messages to be exchanged between the Member States. The NIST file should primarily contain the ten-print fingerprint image and optionally the palm-print images (if available), as grey-scale images of a resolution of 500 dpi, encoded and compressed with the "Wavelet Scalar Quantization" algorithm (WSQ).

Please note that the definition of the detailed content of the NIST file is out of scope of the *ECRIS Technical Specifications* project. It is therefore recommended to apply the same standard for NIST files as the one that has been defined for the PRÜM project. The detailed definition of this standard can be found in the Council Decision 2008/616/JHA of 23 June 2008 on the implementation of Decision 2008/615/JHA on the stepping up of cross-border cooperation, particularly in combating terrorism and cross-border crime, more specifically in "CHAPTER 2: Exchange of dactyloscopic data (interface control document)" of the annex."

The support for such binary attachments is implemented as follows in the technical artefacts:

- The type "BinaryAttachmentType" contains only the meta-data of a binary attachment file; the type "BinaryAttachmentPushType" is an extension of the previous type and contains additionally the binary file itself.

- "Pull" implementation:

  All regular messages that can carry identification information can optionally contain an element of type "BinaryAttachmentType", indicating that fingerprints are available. If a Member State receives such a message, it can use the "deliverRequestAdditionalInfo" operation, using the "BinaryAttachmentID" value provided from the received meta-data as parameter, in order to ask the sending Member State to send the binary file. The Member State having the electronic fingerprints transmits the binary file using the "deliverRequestAdditionalInfoResponsePush" operation, which contains an element of type "BinaryAttachmentPushType".

- "Push" implementation:

  The "push" versions of the messages can optionally contain an element of type "BinaryAttachmentPushType" which immediately carries the fingerprints binary file. These operations are however only to be used if supported by both the sending and the receiving Member State. An implementation that does not support the usage of the "push" operations must systematically throw a *SOAP* fault using the specific fault code "OperationNotSupported".

Please note that the service contract itself cannot enforce the rules relating to the type and size of binary files that are defined in the "Technical Architecture" document, due to the fact that specifications such as *Xmime* or *MTOM-Policy* were discarded for interoperability reasons. These controls are to be implemented in the ECRIS software and appropriate functional error messages are to be returned if an error situation occurs.

The following sequence diagram illustrates the usage of the "pull" services for pulling a binary file:

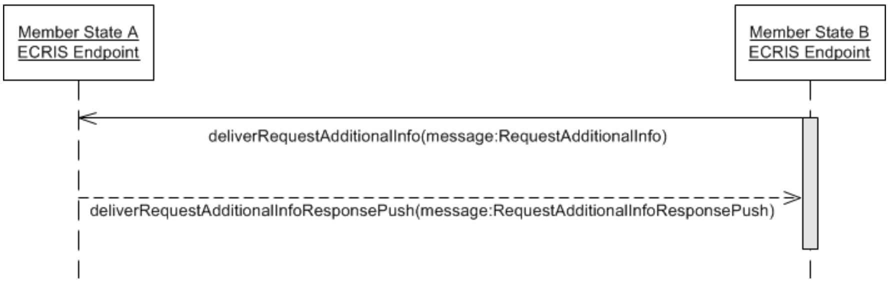

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequestAdditionalInfo(message:RequestAdditionalInfo)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePush)

Figure 29 – Pull implementation kinematic

This sequence can only occur within an on-going request of notification transaction and if the sender of the request or notification message provided information about available fingerprints. In this case, the receiver of the message can ask the sender to provide the NIST file by using the "request additional info" message and asking specifically in this message to retrieve the binary attachment by providing the ID of the binary attachment (i.e. field "MessageDataBinaryIDRequested") and using the value "fingerprints" in the field "MessageDataAdditionalInformationRequested". The sender then pushes the electronic file by using the "push" version of the "response to request for additional info" message.

The following diagram illustrates the pulling of the binary attachment in the context of a request kinematic:

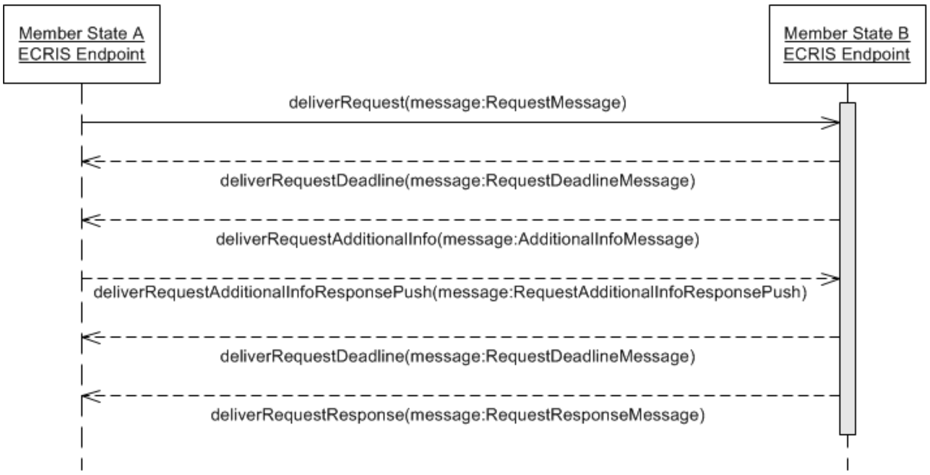

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePush)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 30 – Request kinematic using the pull implementation

4.6 Kinematics

The following chapters illustrate the expected kinematics supported by the ECRIS technical specifications, using UML sequence diagrams. The kinematics described in the following sections cover **all** message exchanges to be supported for ECRIS. They are grouped based on the business process that is supported (i.e. notification or request).

Please note that the more general processing rules for functional errors and cancellation of messages defined in this document must always be kept under consideration and apply to the kinematics presented. Keeping this in mind, the sequence diagrams do not depict all possible combinations of notifications and requests together with functional errors or cancellation messages for the sake of readability.

For the same reason, version-related information such as suffixes in operations or messages has been omitted in the diagrams. The following examples also assume that no technical error occurs resulting in a synchronous SOAP fault to be returned.

4.6.1 Handling Unexpected Messages

As a general rule, an ECRIS message is to be considered **unexpected** if it is not foreseen in any of the kinematics described further in this document. Functional error and cancellation messages are also to be considered **unexpected** in case that the rules defined for handling and sending them are not respected (for example a functional error message being sent as an answer to a request response message).

Depending on the context in which an unexpected message is received, an ECRIS implementation must follow the rules below:

- A message is received outside of any on-going transaction. As an example: Member State "A" has currently no on-going "request" transactions open with Member State "B" but receives from "B" a "request deadline" message. In such a case, the receiver of the unexpected message (1) drops the message, (2) does not reply to the sender automatically and (3) raises an alert to an operator. Indeed, such messages could originate from a security breach and be an attempt of an outside threat to impersonate a Member State.
- A message is received within an on-going transaction but does not correlate to another message previously exchanged in this transaction. In particular, it refers to a message ID (1) that is not known to the receiver or (2) that belongs to another transaction or (3) that does not relate to the previous message but to another one in the transaction. In such cases, the receiver should send back a functional error with the appropriate error code to the sender (i.e. functional error code GEN-2).
- A message is received within an on-going transaction and it does correlate to the previous message but it is not supposed to occur at this moment in the transaction.

Example: request → request deadline → ask for additional ID information → request deadline; in this case the last "request deadline" message is unexpected because no response has been provided yet to the request for additional ID information.

In such cases, the receiver should send back a functional error with the appropriate error code to the sender (i.e. functional error code GEN-3).

4.6.2 Notifications

This chapter illustrates the kinematics supported for the message exchanges in the context of "Notify Convictions and Subsequent Changes" business process. The kinematics described in the following sections cover **all possible and expected combinations** for notifications.

However, as mentioned earlier, a functional error can occur within this transaction at several points as a response to a message. Given that, not all possible functional error combinations are depicted but only a few sample cases are provided.

Deliver notification

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B which processes the notification and sends a notification response back.

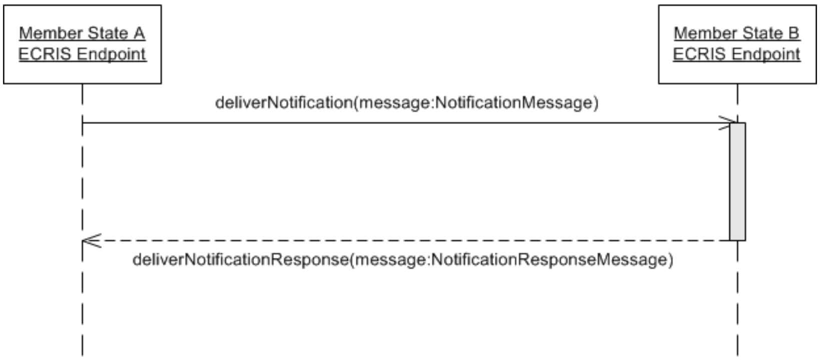

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 31 – "Deliver notification" kinematic

Please note that this kinematic does not support only successful notification message exchanges but also exchanges that result in a business error message. The exact outcome of the processing of the notification is provided by the "MessageDataNotificationResponseTypeReference" property of the "NotificationResponseMessage" element.

In any case, the notification response closes the transaction.

Deliver notification («Push» version)

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. All other mechanics and rules already described earlier remain the same.

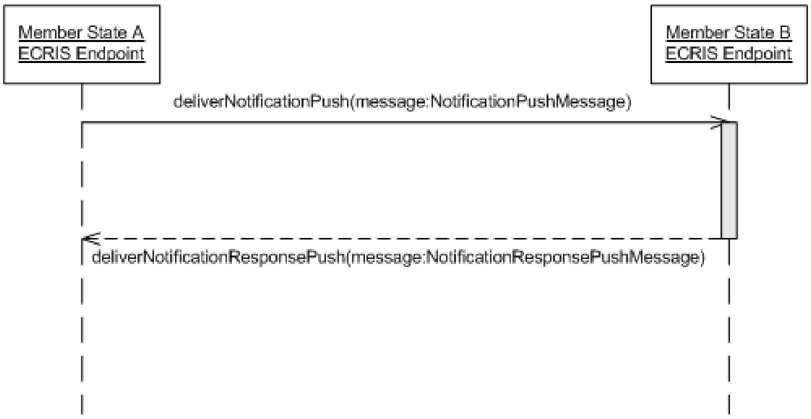

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotificationPush(message:NotificationPushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 32 – "Deliver notification ("push" version) kinematic

Deliver notification > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B which encounters a functional error when processing the notification. Member State B sends a functional error message back to Member State A.

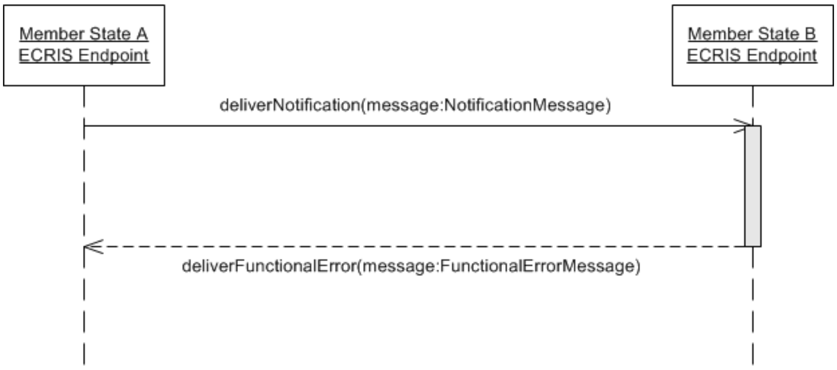

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverFunctionalError(message:FunctionalErrorMessage)

Figure 33 – "Deliver notification > functional error" kinematic

Please note that, as already explained earlier, in this scenario the functional error response also closes the transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the notification message (i.e. "deliverNotificationPush" operation).

Deliver notification > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. Shortly afterwards, an operator within the central authority of Member State A realises a significant problem in the information transmitted. Therefore, he decides to issue a cancellation message for annulling the notification already sent.

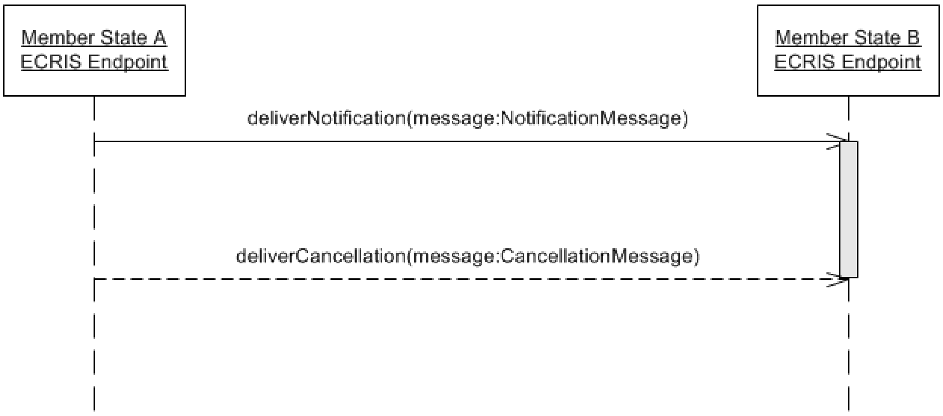

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverCancellation(message:CancellationMessage)

Figure 34 – "Deliver notification > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message also closes the transaction. The operator of Member State A can then correct the problem and if required send the corrected notification later, which will initialise a new transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the notification message (i.e. "deliverNotificationPush" operation).

Deliver notification > notification response > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. Member State B sends the notification response but shortly afterwards, an operator within the central authority of Member State B realises a significant problem in the response transmitted. Therefore, he decides to issue a cancellation message for annulling the notification response already sent.

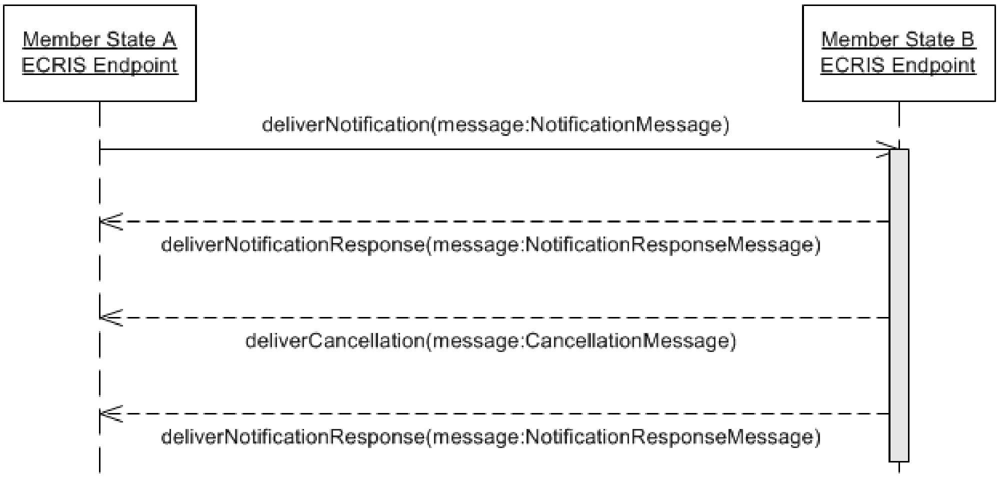

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)
deliverCancellation(message:CancellationMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 35 – "Deliver notification > notification response > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message reopens the previously closed transaction. The central authority of Member State B is expected to correct the problem and to send a corrected notification response later, which will again close the transaction.

Please note also that this kinematic is identical for the "push" version of the notification messages (i.e. "deliverNotificationPush" and "deliverNotificationResponsePush" operations).

Deliver notification > request additional ID info

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B sends a notification response back to Member State A.

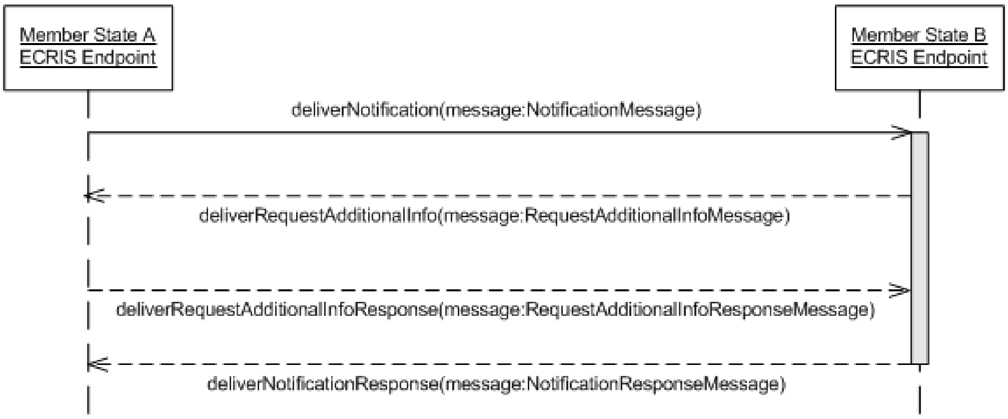

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverRequestAdditionalInfo(message:RequestAdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:RequestAdditionalInfoResponseMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 36 – "Deliver notification > request additional ID info" kinematic

Please note that this kinematic does not support only the case where additional information is available but also the case where such information is unavailable. The exact outcome of the processing of the request for additional identification information is provided by the "MessageDataAdditionalInfoResponseTypeReference" property of the "RequestAdditionalInfoResponseMessage" element.

Also, a Member State receiving a notification can, in theory, issue requests for additional information multiple times during the transaction. This case is very similar to the "deliver notification > request additional info" kinematic but is not further elaborated in this document. It should however be supported by the ECRIS implementations, following the principle described here.

In any case, the notification response closes the transaction.

Please note also that the fact that additional identification information is received does not have any impact on the notification's deadline (i.e. the operational deadline as defined in the "Business Analysis" document). In particular, no new deadline is calculated and sent by the notified Member State to the notifying Member Sate.

Deliver notification («Push» version) > request additional ID info

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics and rules already described earlier remain the same.

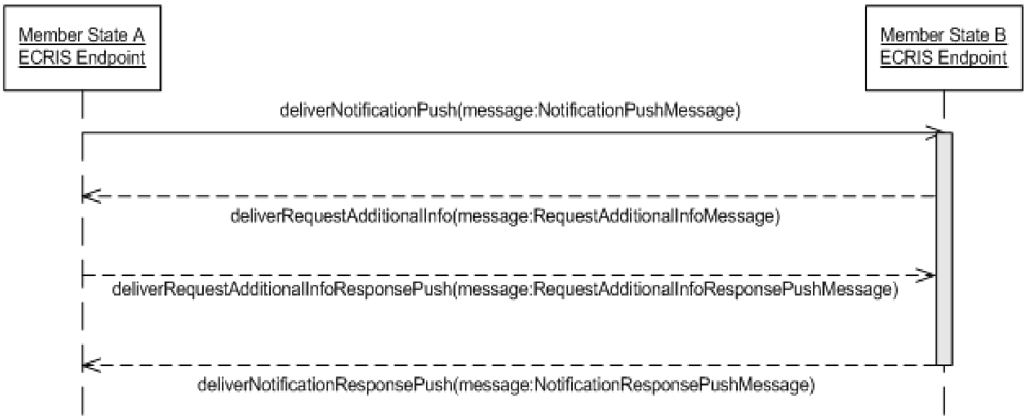

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverNotificationPush(message:NotificationPushMessage)
deliverRequestAdditionalInfo(message:RequestAdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:RequestAdditionalInfoResponsePushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 37 – "Deliver notification ("push" version) > request additional ID info" kinematic

Deliver notification > request additional ID info > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a notification to Member State B. During the processing of the notification, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B encounters a functional error when processing the additional identification information transmitted by Member State A. Member State B sends a functional error message back to Member State A. Member State A sends a corrected version of the additional identification information to Member State B. Finally Member State B sends a notification response back to Member State A.

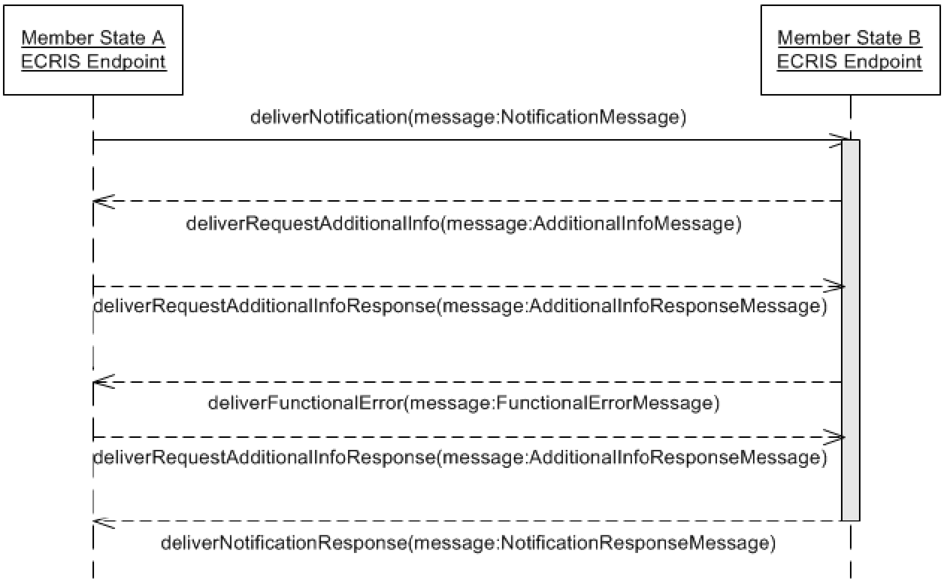

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotification(message:NotificationMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverNotificationResponse(message:NotificationResponseMessage)

Figure 38 – "Deliver notification > request additional ID info > functional error" kinematic

Please note that in this case the functional error message does **not** close the transaction as the Member State providing additional information can make the appropriate corrections and issue again the corrected information. Please also note that the "deliverFunctionalError" operation can be invoked at any point **after** the "deliverNotification" operation and **before** the "deliverNotificationResponse" operation are invoked.

Deliver notification («Push» version) > request additional ID info > functional error occurs

This scenario is similar to the previous one with the exception that the process is started with the "deliverNotificationPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics and rules already described earlier remain the same.

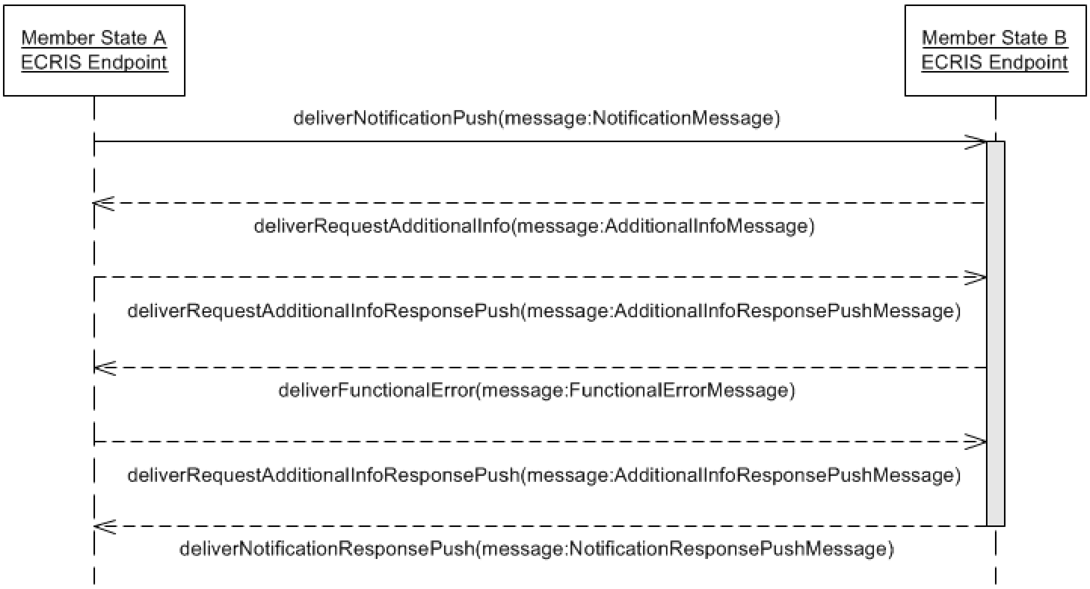

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverNotificationPush(message:NotificationMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverNotificationResponsePush(message:NotificationResponsePushMessage)

Figure 39 – "Deliver notification ("push" version) > request additional ID info > functional error" kinematic

4.6.3 Requests

This chapter illustrates the kinematics supported for the message exchanges in the context of the "Request Criminal Record Information" business process. The kinematics described in the following sections cover **all possible and expected combinations** for requests.

However, as mentioned earlier, a functional error can occur within this transaction at several points as a response to a message. Given that, not all possible functional error combinations are depicted but only a few sample cases are provided.

Deliver request

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline to Member State A, processes the request and sends a request response back.

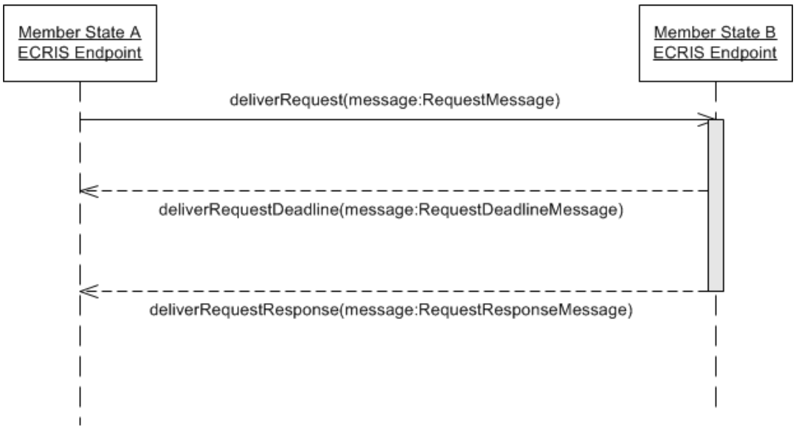

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 40 – "Deliver request" kinematic

Please note that this kinematic does not support only successful request message exchanges but also exchanges that result in a business error message during the processing or in a request denial. The exact outcome of the processing of the request is provided by the "MessageDataRequestResponseTypeReference" property of the "RequestResponseMessage" element.

In any case, the request response closes the transaction.

Deliver request («Push» version)

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. All other mechanics and rules already described earlier remain the same.

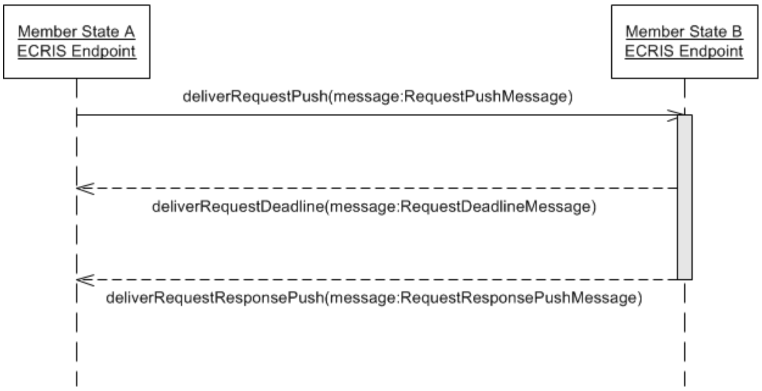

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequestPush(message:RequestPushMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponsePush(message:RequestResponsePushMessage)

Figure 41 – "Deliver request ("push" version)" kinematic

Deliver request > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which encounters a functional error when processing the request message. Member State B sends a functional error message back to Member State A.

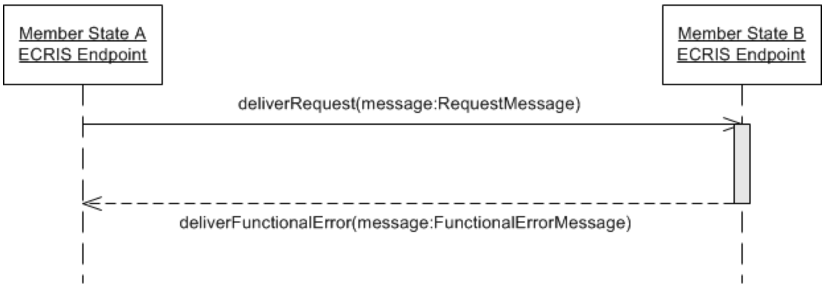

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverFunctionalError(message:FunctionalErrorMessage)

Figure 42 – "Deliver request > functional error" kinematic

Please note that, as already explained earlier, in this scenario the functional error response also closes the transaction.

Please note also that this kinematic is identical if the transaction starts with the "push" version of the request message (i.e. "deliverRequestPush" operation).

Deliver request > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B. Member State B receives the request and sends to Member State A the deadline for replying the request. At some point, before the deadline is expired or a reply is issued, the central authority in Member State A that authored the request realises that a mistake was made in this request. At that point, it issues a cancellation message for annulling the request.

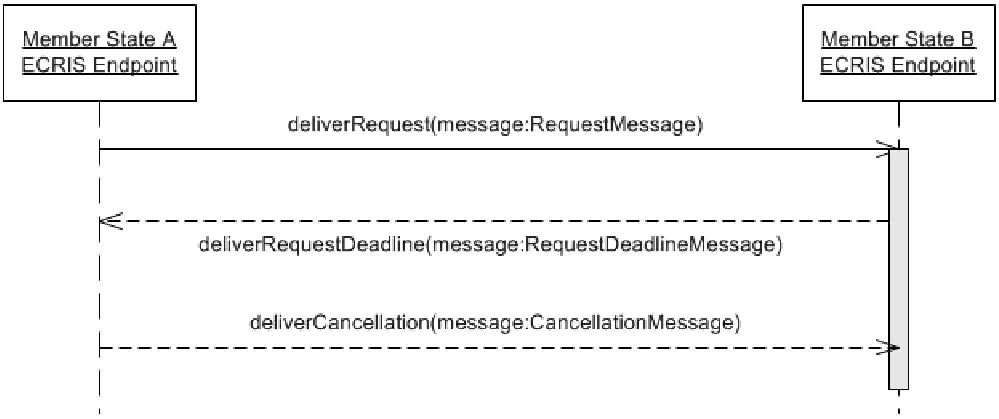

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverCancellation(message:CancellationMessage)

Figure 43 – "Deliver request > cancellation" kinematic

As mentioned earlier, the cancellation message for a request essentially terminates the message exchange. Thus, Member State B has to consider the transaction closed and stop any further actions so as to satisfy the request.

Please note that it is recommended to fully automate the calculation and sending of the legal deadline for requests. Therefore the diagram above illustrates the expected sequence of operations. However, the cancellation can also theoretically occur before sending the deadline for the request and must also be supported as a valid kinematic, as depicted below:

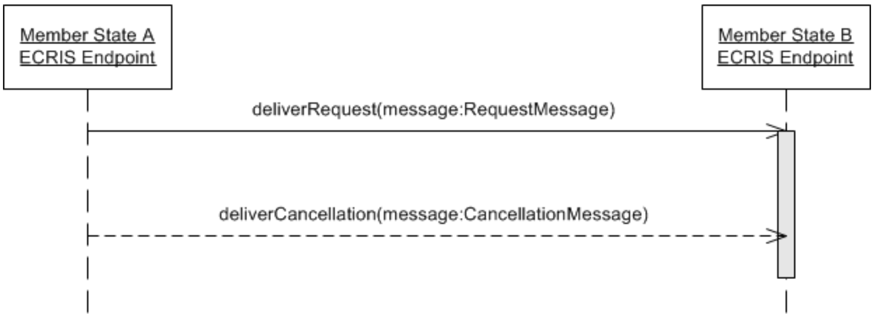

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverCancellation(message:CancellationMessage)

Figure 44 – "Deliver request > cancellation" kinematic

Deliver request > request response > cancellation

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B. Member State B transmits the legal deadline and the response to the request but shortly afterwards, an operator within the central authority of Member State B realises a significant problem in the response transmitted. Therefore, he decides to issue a cancellation message for annulling the request response already sent.

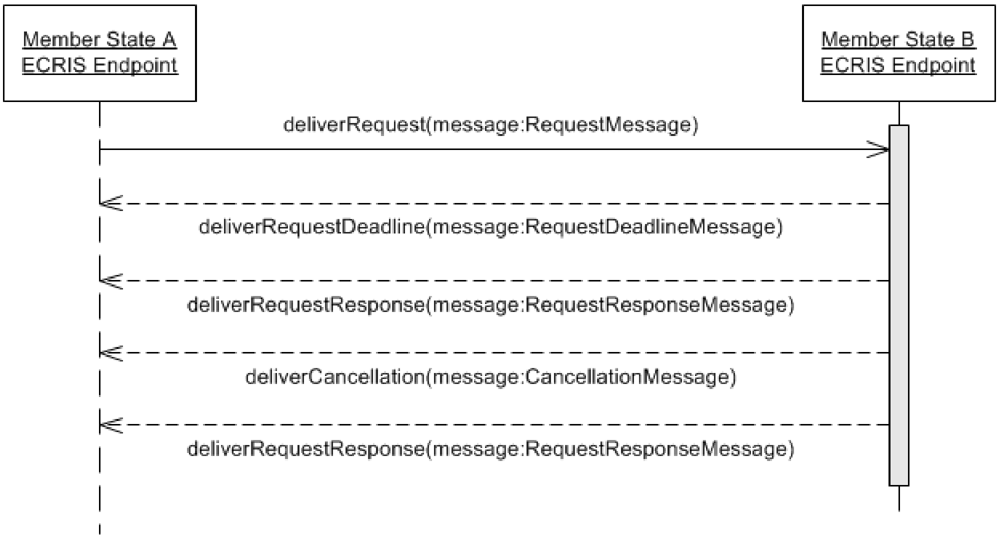

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)
deliverCancellation(message:CancellationMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 45 – "Deliver request > request response > cancellation" kinematic

Please note that, as already explained earlier, in this scenario the cancellation message reopens the previously closed transaction. The central authority of Member State B is expected to correct the problem and to send a corrected request response later, which will again close the transaction.

This kinematic is identical for the "push" version of the request messages (i.e. "deliverRequestPush" and "deliverRequestResponsePush" operations).

Deliver request > request additional ID info

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline back to Member State A. During the processing of the ~~notification~~ request, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B sends a request response back to Member State A.

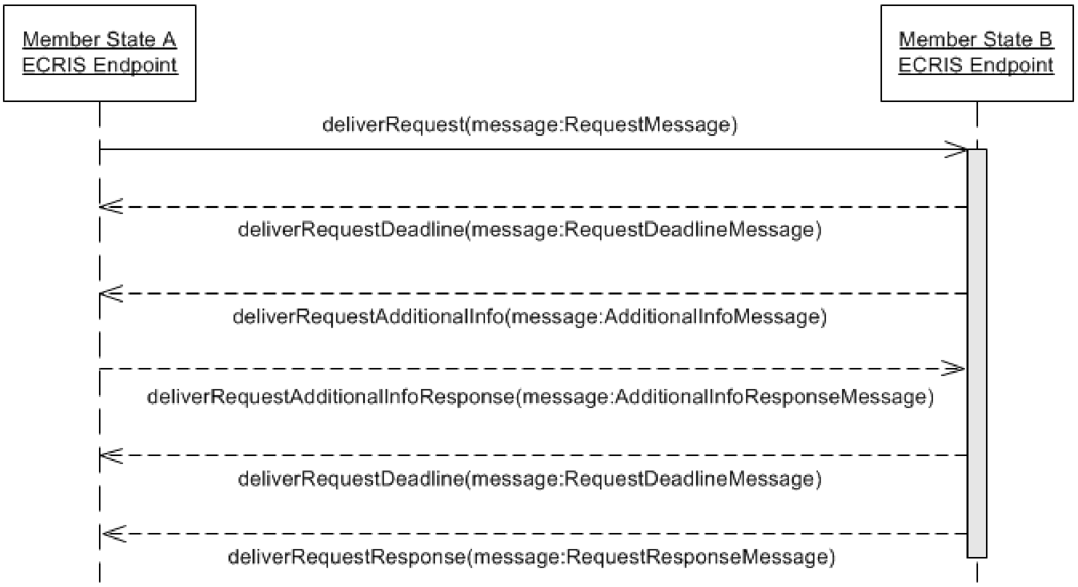

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 46 – "Deliver request > request additional ID info" kinematic

Please note that this kinematic does not support only the case where additional information is available but also the case where such information is unavailable. The exact outcome of the processing of the request for additional identification information is provided by the "MessageDataAdditionalInfoResponseTypeReference" property of the "RequestAdditionalInfoResponseMessage" element.

Also, a Member State receiving a request can, in theory, issue requests for additional information multiple times during the transaction. This case is very similar to the "deliver request > request additional info" kinematic but is not further elaborated in this document. It should however be supported by the ECRIS implementations, following the principle described here.

Here also, the diagram above illustrates the expected sequence of operations. However, the request for additional information can also theoretically occur before sending the deadline for the request and must also be supported as a valid kinematic, as depicted below:

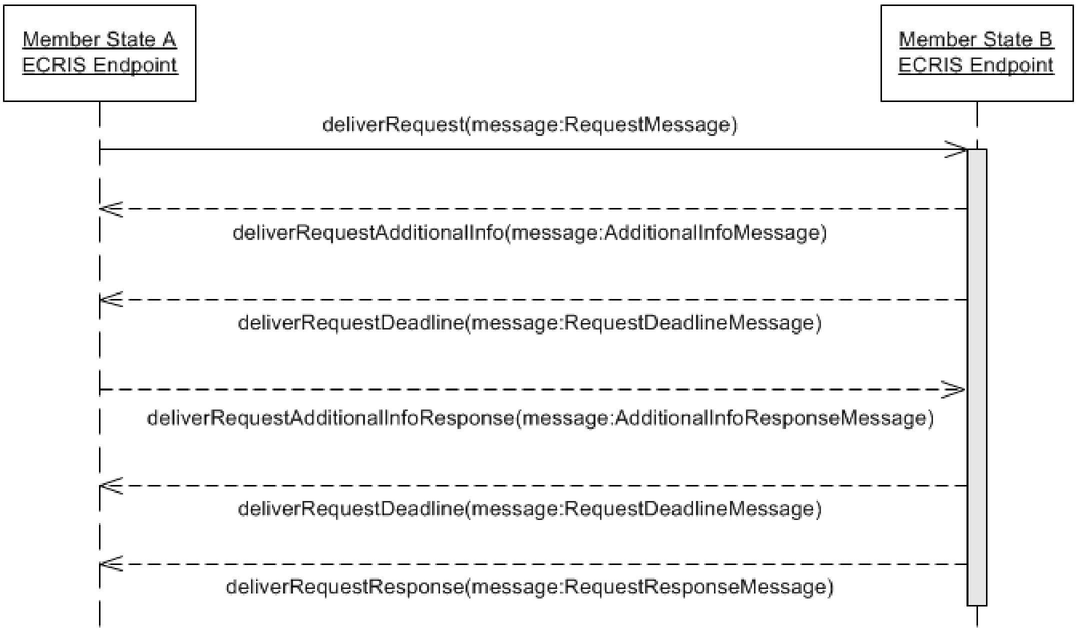

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 47 – "Deliver request > request additional ID info" kinematic

In any case, the request response closes the transaction.

Deliver request («Push» version) > request additional ID info

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation.

All other mechanics, rules and variations already described earlier remain the same. In particular here also the request for additional information can theoretically occur before sending the deadline for the request.

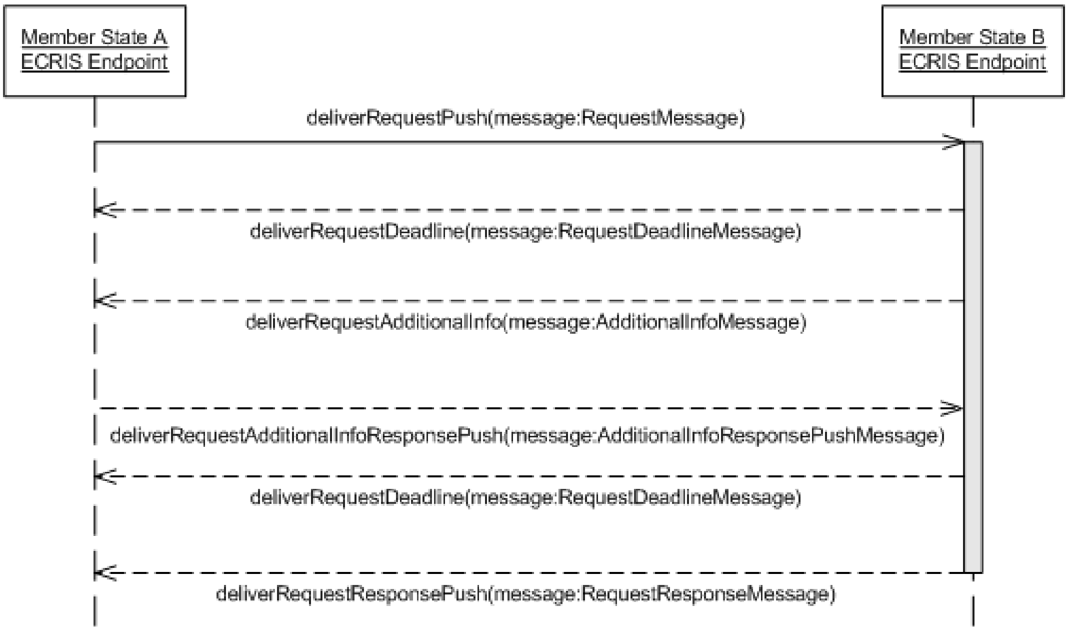

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequestPush(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponsePush(message:RequestResponseMessage)

Figure 48 – "Deliver request ("push" version) > request additional ID info" kinematic

Deliver request > request additional ID info > functional error occurs

This sequence diagram illustrates the following scenario:

Member State A sends a request to Member State B which transmits the legal deadline back to Member State A. During the processing of the ~~notification~~ request, Member State B requests additional identification information to Member State A. Member State A provides the additional information requested. Member State B encounters a functional error when processing the additional identification information transmitted by Member State A. Member State B sends a functional error message back to Member State A. Member State A sends a corrected version of the additional identification information to Member State B. Finally, Member State B sends a request response back to Member State A.

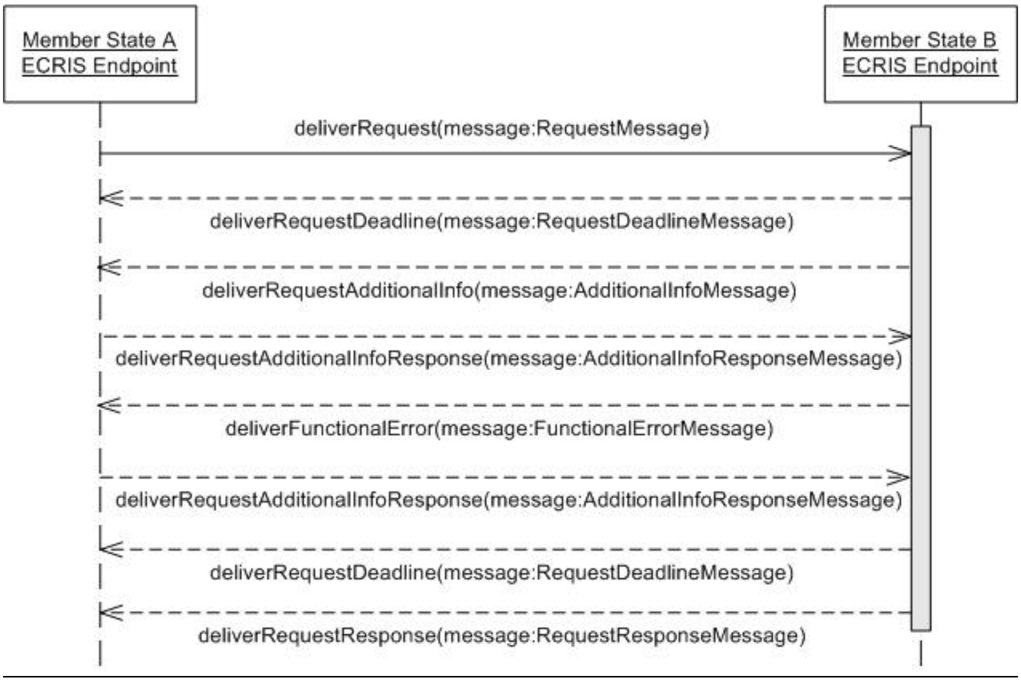

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 49 – "Deliver request > request additional ID info > functional error" kinematic

Please note that in this case the functional error message does **not** close the transaction as the Member State providing additional information can make the appropriate corrections and issue again the corrected information. Please also note that the "deliverFunctionalError" operation can be invoked at any point **after** the "deliverRequest" operation and **before** the "deliverRequestResponse" operation are invoked.

Here also, the diagram above illustrates the expected sequence of operations. However, the sending of the new deadline for the request can theoretically also occur before sending the functional error and must also be supported as a valid kinematic, as depicted below:

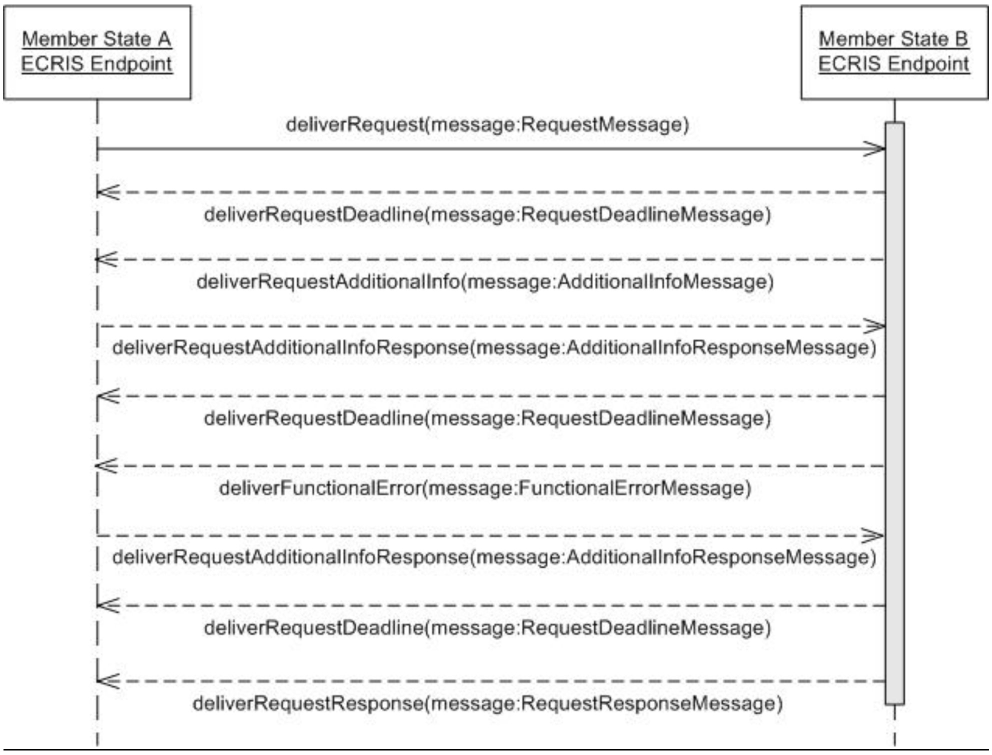

Member State A ECRIS Endpoint
Member State B ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 50 – "Deliver request > request additional ID info > functional error" kinematic

Given the variations that are also possible for sequence of occurrence of the first deadline message, the following diagrams also illustrate valid kinematics that need to be supported by the ECRIS software:

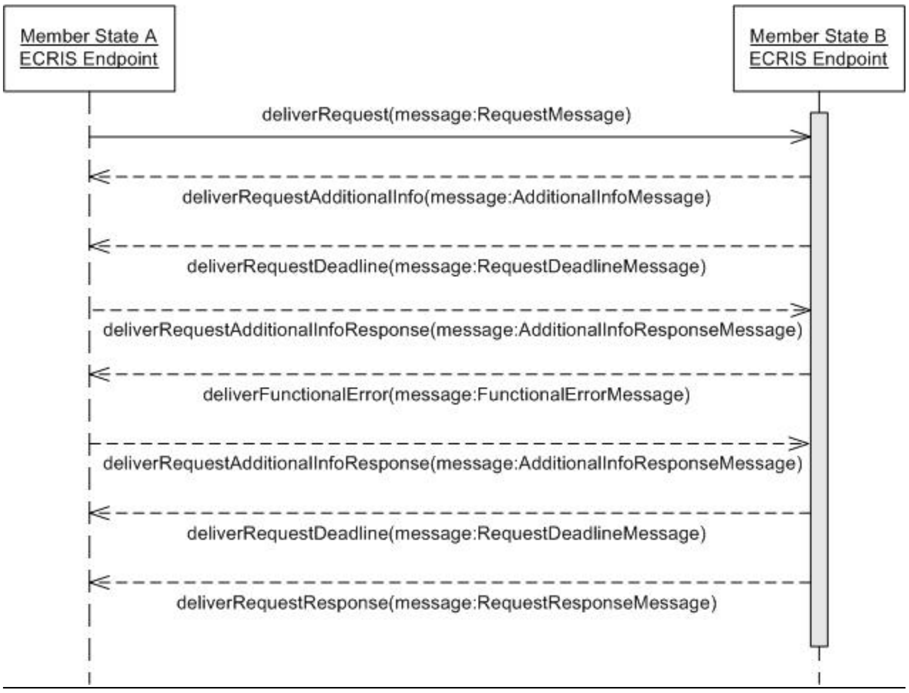

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 51 – "Deliver request > request additional ID info > functional error" kinematic

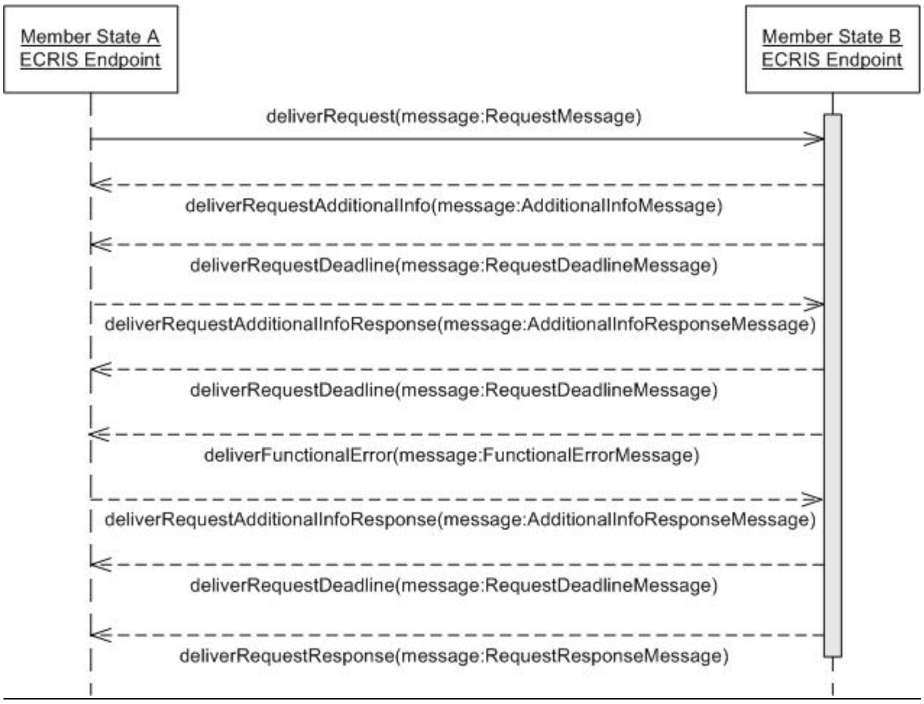

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequest(message:RequestMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponse(message:AdditionalInfoResponseMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestResponse(message:RequestResponseMessage)

Figure 52 – "Deliver request > request additional ID info > functional error" kinematic

In any case, the request response closes the transaction.

Deliver request («Push» version) > request additional ID info > functional error occurs

This scenario is similar to the previous one with the exception that the process is started with the "deliverRequestPush" operation which carries the fingerprints binary file. Since both Member States implement the "push" approach, the additional identification information is also transmitted using the "push" operation. All other variations described in the previous request kinematics also apply to the "push" version of this message exchange.

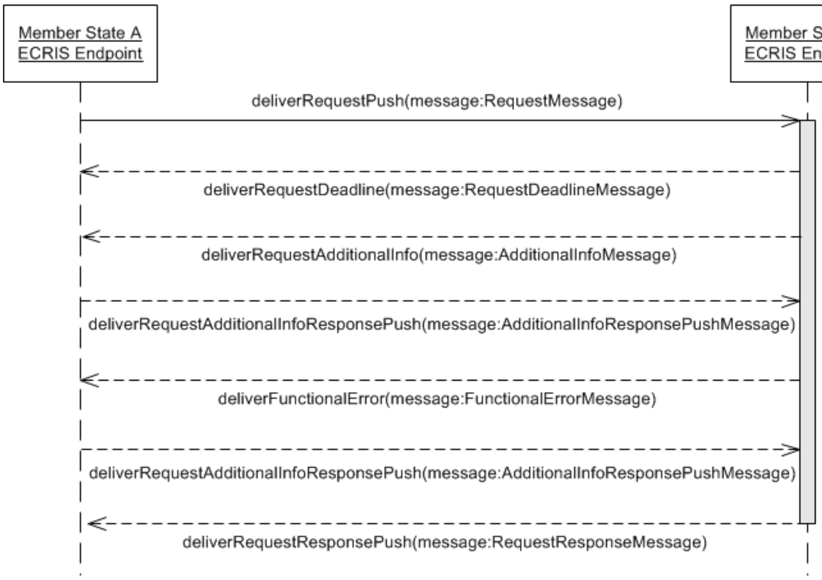

Member State A
ECRIS Endpoint
Member State B
ECRIS Endpoint
deliverRequestPush(message:RequestMessage)
deliverRequestDeadline(message:RequestDeadlineMessage)
deliverRequestAdditionalInfo(message:AdditionalInfoMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverFunctionalError(message:FunctionalErrorMessage)
deliverRequestAdditionalInfoResponsePush(message:AdditionalInfoResponsePushMessage)
deliverRequestResponsePush(message:RequestResponseMessage)

Figure 53 – "Deliver request ("push" version) > request additional ID info > functional error" kinematic

5 IMPLEMENTATION

This chapter provides rules, guidelines and recommendations to the implementers of the detailed technical specifications of ECRIS.

The rules defined below have to be followed **strictly** by all software implementations so as to ensure the proper operation of the software and effective data exchanges between Member States.

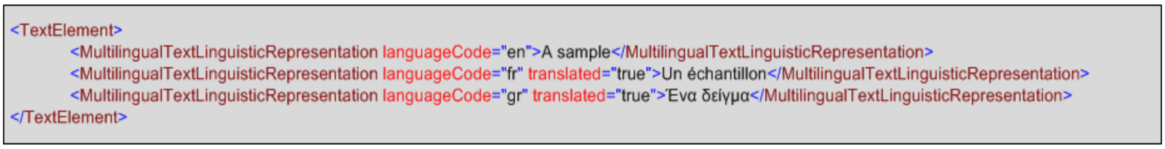
The guidelines and recommendations described further below are based on:

- Practical experience from the NJR pilot project
- Lessons learned gathered during the design and validation of the technical artefacts
- Best practices of software integration

5.1 Transmitting Text in Several Languages

Given its nature, ECRIS needs to support transmission of translated ~~and/or transliterated~~ text. To that end, a specific structure called "MultilingualTextType" is defined in the domain model. This type provides the necessary facilities for sending the same text value in various languages. It also allows indicating whether a text value is the result of a translation ~~and/or transliteration~~.

This functionality is achieved by defining a property with unbounded cardinality for "MultilingualTextType", called "MultilingualTextLinguisticRepresentation". This property can take any text value and offers a set of properties for designating whether the text is translated ~~or transliterated~~. In the following sample, a "TextElement" of type "MultilingualTextType" is used to transmit the text value "A sample" in three different languages.



```
<TextElement>
        <MultilingualTextLinguisticRepresentation languageCode="en">A sample</MultilingualTextLinguisticRepresentation>
        <MultilingualTextLinguisticRepresentation languageCode="fr" translated="true">Un échantillon</MultilingualTextLinguisticRepresentation>
        <MultilingualTextLinguisticRepresentation languageCode="gr" translated="true">Ένα δείγμα</MultilingualTextLinguisticRepresentation>
</TextElement>
```

Figure 54 – A sample of using "MultilingualTextType"

There are at least two use cases that have been identified and that would require the usage of such multi-lingual text representation:

- By the ECRIS legal basis, request messages have to be sent in the language of the requested Member State. ~~In the "Business Analysis" it has been defined to also be able to send the request additionally in the language of the sender.~~ With the proposed solution, the same request message can now contain free text elements in ~~both~~ multiple languages at the same time.
- Request responses can be constituted of information extracted from the national register to which foreign convictions that have been stored outside of the register for the sole purpose of retransmission are added. In such cases, the response to a request can contain text in multiple languages. Also, for the foreign convictions that were stored in the national register, it may be interesting to be able to send the texts that were translated in both their original language and in the translated language.

5.2 Correlating Convictions

As defined in the "Business Analysis" document, it is required to be able to establish relations with previously transmitted convictions in notification messages for two specific situations:

- When a notification message provides an update of a conviction previously notified, the notification message needs to refer to the conviction that is actually being modified.

- In the case of the formation of an overall sanction, a notification on this new conviction is sent indicating the formation of the overall sanction. This notification message indicates that this conviction affects other convictions previously notified by referring to them.

Considering the fact that an update of a conviction may occur for convictions that have been notified to the Member State of nationality before NJR or before ECRIS, the notification message must be able to relate to a previously notified conviction by using one of the following:

- For convictions that have already been notified using ECRIS: the unique ECRIS identifier of the conviction, or
- For convictions that have been notified using NJR: the unique NJR identifier of the conviction (called "decision id"), or
- For convictions that have not been notified before or that have been notified by other means of communication: the code and name of the convicting authority, the file number of the conviction and the final date of the conviction. The code of the convicting authority is defined as optional in this structure due to the fact that not all Member States assign such codes.

In order to fulfil these requirements, the properties "MessageDataUpdatedConviction" and "MessageDataOtherAffectedConviction" have been defined as part of notification messages (i.e. in the "NotificationMessageDataType"). These allow establishing a relation with previous convictions that have either been transmitted via ECRIS, via NJR or via other means of communication.

The property "MessageDataUpdatedConviction" is to be used for indicating that the notification provides an update of the conviction being referred to.

The property "MessageDataOtherAffectedConviction" is to be used for indicating that the conviction contained in the notification message affects other convictions such as in the case of the formation of overall sanction.

Please note that the rules and structures defined in these technical specifications imply that a unique ECRIS technical identifier will need to be assigned to each conviction the first time that it is transmitted through ECRIS in a notification message. This applies also to convictions that may have previously been transmitted using other types of technical identifiers through NJR or any other means of communication.

5.3 XSI:TYPE

In the XSD type definitions of the detailed technical specifications, the "xsi:type" attribute is not explicitly defined. It must be noted however that software implementations generating the XML documents from in-memory objects however explicitly generate the "xsi:type" attribute in complex elements so as to know how to serialise/de-serialise properly from in-memory object to XML and vice versa (also called marshalling/un-marshalling).

It must be noted here also that if sample XML is generated manually from the XSD definitions using tools like SoapUI, the "xsi:type" attribute is not included which may cause problems with some implementations which will not be able to detect into which type of object the XML should be de-serialised.

As a more general rule, the usage of the "xsi:null" attribute should be avoided in the ECRIS implementations so as to avoid interoperability issues.

5.4 Technical Validation Rules

Before describing the technical validation rules, please note that implementers **must** first check whether their specific platform properly implements the "SOAP Version 1.2 Part 2: Adjuncts" and in particular chapter §7.5.1 "Behaviour of Requesting SOAP Node". This is important due to the fact that different platforms interpret in a different way the specification.

Taking this under consideration, it is **expected** that an implementation is capable of returning an HTTP status code 500 whenever a fault is returned to the sender.

The following conditions are to be implemented by the ECRIS software and **must** result in a **synchronous technical error** when encountered:

- The message payload is not structurally valid (i.e.it does not validate against the *XSD*)
- The version of a web service operation used by the sender is no longer valid (please refer to the detailed explanation in the "Technical Architecture" document). This must result in a SOAP fault with the code "WrongVersionUsed".
- The operation invoked is not supported. This situation can **only** occur when a sender is invoking a "Push" operation on a receiver that does not support such an operation mode. This must result in a SOAP fault with the code "OperationNotSupported".

Please note that in addition to the error conditions mentioned above which are specific to ECRIS, various other situations resulting in an HTTP 500 status code and a SOAP fault as return message can occur. As a consequence, any given application server or *web service* framework can potentially return such an HTTP 500 status code for various other reasons such as connectivity problems, hardware failure, etc.. It is not in the scope of this document to provide an exhaustive list of all such possible causes.

Additionally, the HTTP client software **must** always perform the required security checks defined in the "Security Analysis" document for validating the HTTPS certificate provided from the requested server. In case any of those checks fails, the client software **must** immediately drop the connection and issue appropriate warnings to the operators of the ECRIS software so that manual action can be taken.

In any case, technical errors need to be logged thoroughly. It is also recommended that implementers communicate their findings to the central entity so as to constitute a technical knowledge base that can be shared with all Member States experts.

5.5 Functional Validation Rules

The following sections define the conditions that **can** lead to **returning a functional error** to the sender of the message when encountered.

First of all, two categories of functional validation conditions need to be distinguished:

- "automatic" functional validation – the conditions can easily be verified in an automated manner by the ECRIS software which then can automatically send the appropriate error code to the sender of the message.
- "manual" functional validation – the conditions can only be verified by human operators; for these cases, the detailed technical specifications only define appropriate error codes that can be used by the Member States' central authorities in case where this error situation is identified when processing the message during the transaction.

It is left at the discretion of each implementer to decide if and in what manner the validation processes are implemented. However each ECRIS software implementation must be capable of at least receiving and understanding the functional error codes defined.

In order to minimise the occurrence of functional errors, it is also recommended that the ECRIS software of the sending Member State applies the functional validation rules described in the next sections before transmitting the message.

For each error condition described in the next sections, the common reference table "FunctionalErrors" defines a corresponding error code to be used in the functional error messages to be returned to the sender of a problematic ECRIS message.

5.5.1 « Automatic » Validation

The evaluation of the following conditions can be automated in the ECRIS software and should lead to a functional error when encountered in an ECRIS message.

General

[GEN-1] The size of the message, including attachments, exceeds 10 megabytes.

(Please note that this size limit for all ECRIS messages has been defined and agreed upon in the "Technical Architecture" document so as to limit the overall volume and network bandwidth to be consumed by the ECRIS data exchanges.)

[GEN-2] One of the following messages does not correlate to a previously received message, meaning that the technical identifier indicated in the property "MessageMetaDataResponseTo" does not match the technical identifier of any previously received message:

- CancellationMessage
- FunctionalErrorMessage
- RequestDeadlineMessage
- RequestResponseMessage
- RequestResponsePushMessage
- NotificationResponseMessage
- NotificationResponsePushMessage
- RequestAdditionalInfoMessage
- RequestAdditionalInfoResponseMessage
- RequestAdditionalInfoResponsePushMessage

[GEN-3] One of the following messages does correlate to a previously received message within the same transaction but is not expected to occur at this moment of the transaction, according to the defined kinematics:

- CancellationMessage
- RequestDeadlineMessage
- RequestResponseMessage
- RequestResponsePushMessage
- RequestAdditionalInfoMessage
- RequestAdditionalInfoResponseMessage
- RequestAdditionalInfoResponsePushMessage
- NotificationResponseMessage
- NotificationResponsePushMessage

Binary Attachments

[BIN-1] The binary file attached is not a NIST file.

[BIN-2] The binary file attached is corrupt or has a different MD5 hash code than the one provided.

"Person" entity

[PER-1] The "surname" property of the person contains a dummy value (i.e. element `dm:PersonType:PersonName:Surname`).

[PER-2] The birth date of the person is equal or after the current date.

[PER-3] The birth date of the alias of the person is equal or after the current date.

[PER-4] The birth date of the person is after the issuing date of the person's identification document.

[PER-5] The birth date of the alias of the person is after the issuing date of the identification document of the alias of the person.

[PER-6] The alias fore- and surname contain identical information as the person fore- and surname.

[PER-7] The birth date is equal or after any of the following dates:

- Decision: decision date
- Decision: final decision date
- Offence: start date
- Offence: end date
- Sanction: execution start date
- Sanction: execution end date
- Sanction: sentenced start date
- Sanction: sentenced end date

"Place" entity

[PLA-1] The town code references a value in the common reference table "Cities" that is in a country that is different than the "country" property of the entity.

[PLA-2] The country subdivision references a value in the common reference table "Country Subdivisions" that is in a country that is different than the "country" property of the entity.

Please note that if the "town name" property does not match the value referenced by the "town code" property, it is assumed that the ~~"town name"~~ "town code" has precedence. No functional error is to be returned in this case.

Please note also that if the town name provided is misspelled or spelled in another language than the one expected (e.g. indicating "Saragosa" instead of "Zaragoza", "Parigi" instead of Paris, etc.), it should not lead to an automatic rejection of the message carrying the information. Here also, no functional error is to be returned in this case.

"Identification Document" entity

[IDD-1] The issuing date is after the "valid until" date.

[IDD-2] The issuing date is after the current date.

"Decision" entity

[DEC-1] The decision date is after the final decision date.

[DEC-2] The decision date is after the current date.

[DEC-3] The final decision date is after the current date.

[DEC-4] The decision date is before the decision date of the initial conviction.

[DEC-5] The decision is not related to any sanction within the message.

Please note that this error is to be returned in case a notification message contains a "decision" entity without any relation to at least one "sanction" entity within the same message. This applies only if the field "change type" contains one of the following values:

- "a – Suspended penalty/measure"
- "b – Partially suspended penalty/measure"
- "c - Suspended penalty/measure with probation/supervision"
- "d - Partially suspended penalty/measure with probation/supervision"
- "e – Conversion of penalty/measure"
- "h – Revocation of suspended penalty/measure"
- "i - Subsequent formation of an overall penalty"
- "j - Interruption of enforcement/postponement of the penalty/measure"
- "k - Remission of the penalty"
- "l - Remission of the suspended penalty"
- "n - End of penalty"

If the field "change type" is left empty or if it has another value than the ones listed above, this error is not to be returned.

"Conviction" entity

[CON-1] The field "non-criminal ruling" is set to "true" and one of the sanctions in this conviction is in one of the following common category (including sub-categories):

- "1000 – Deprivation of freedom"
- "4000 – Prohibition or expulsion from territory"
- "10000 – Military penalty"

[CON-2] The field "non-criminal ruling" is set to "true" and one of the offences in this conviction is in one of the following common category (including sub-categories):

- "0100 00 Crimes within the jurisdiction of the International Criminal Court",
- "0200 00 Participation in a criminal organisation"
- "0300 00 Terrorism"

- “0400 00 Trafficking in human beings”
- “0500 00 Illicit trafficking and other offences related to weapons, firearms, their parts and components, ammunition and explosives”
- “1400 00 Tax and customs offences”
- “1900 00 Forgery of means of payment

[CON-3] The conviction’s decision date is after the conviction’s final decision date.

[CON-4] The conviction’s decision date is after the current date.

[CON-5] The conviction’s final decision date is after the current date.

“Offence” entity

[OFF-1] The end date is before the start date.

[OFF-2] The start date is after the decision date of the conviction.

[OFF-3] The start date is after the final decision date of the conviction.

[OFF-4] The end date is after the final decision date of the conviction.

[OFF-5] The start date is equal or after the current date.

[OFF-6] The end date is equal or after the current date.

[OFF-7] The common category references a valid value in the common reference table while the national title indicates the dummy value “UNKNOWN”.

[OFF-8] The common category and national title reference dummy values while at least one of the following fields contains a non-dummy value:

- Number of occurrences
- Level of completion
- Level of participation
- Applicable legal provisions
- Place
- Recidivism

“Sanction” entity

[SAN-1] The common category references a valid value in the common reference table while the national title indicates the dummy value “UNKNOWN”.

[SAN-2] The number of fines indicates a value > 0 while the sanction category does not reference the category “8000 - Financial Penalty category code” or one of its child categories.

[SAN-3] The sanction category indicates “8000 - Financial Penalty category code” or one of its child categories but the amount of individual fine is not provided (i.e. the XML element indicating the amount of the individual fine is either not present or its value is set to 0).

[SAN-4] The number of fines indicates a value > 0 while the amount of individual fine is not provided.

[SAN-5] The amount of individual fine indicates a value > 0 while the sanction category does not reference the category “8000 - Financial Penalty category code” or one of its child categories.

[SAN-6] The currency of fine references a valid value while both the number of fines and the amount of individual fine are not provided.

[SAN-7] The currency of fine references a valid value while the sanction category does not reference the category “8000 - Financial Penalty category code” or one of its child categories.

[SAN-8] The number of fines and the amount of individual fine contain valid values but the currency of fine is not provided.

[SAN-9] The sanction category indicates “1001 – Imprisonment” but the sentenced duration is not provided (i.e. the XML element indicating the sentenced duration is either not present or its value is set to 0).

[SAN-10] The execution start date is before the decision date of the conviction.

[SAN-11] The execution start date is before the final decision date of the conviction.

[SAN-12] The execution end date is before the decision date of the conviction.

[SAN-13] The execution end date is before the final decision date of the conviction.

[SAN-14] The execution start date is equal or before the offence start date.

[SAN-15] The execution start date is before the offence end date.

[SAN-16] The execution end date is equal or before the offence start date.

[SAN-17] The execution end date is equal or before the offence end date.

[SAN-18] The execution start date is after the execution end date.

[SAN-19] The execution end date is earlier than (execution start date + duration).

[SAN-20] The sentenced start date is before the decision date of the conviction.

[SAN-21] The sentenced start date is before the final decision date of the conviction.

[SAN-22] The sentenced end date is before the decision date of the conviction.

[SAN-23] The sentenced end date is before the final decision date of the conviction.

[SAN-24] The sentenced start date is equal or before the offence start date.

[SAN-25] The sentenced start date is before the offence end date.

[SAN-26] The sentenced end date is equal or before the offence start date.

[SAN-27] The sentenced end date is equal or before the offence end date.

[SAN-28] The sentenced start date is after the sentenced end date.

[SAN-29] The sentenced end date is earlier than (sentenced start date + duration).

[SAN-30] The common category and the national title reference dummy values while at least one of the following fields contains a non-dummy value:

- Alternative
- Specific to minor
- Execution start date
- Execution end date
- Execution duration
- Number of fines
- Amount of individual fine
- Currency of fine
- Suspension
- Interruption

“Suspension” entity

[SUS-1] The end date is before the start date.

[SUS-2] The start date is before the decision date of the conviction.

[SUS-3] The start date is before the final decision date of the conviction.

[SUS-4] The start date is before the start date of the offence.

[SUS-5] The start date is before the end date of the offence.

[SUS-6] The end date is before the decision date of the conviction.

[SUS-7] The end date is before the final decision date of the conviction.

[SUS-8] The end date is before the start date of the offence.

[SUS-9] The end date is before the end date of the offence.

[SUS-10] The end date is before the execution start date of the sanction.

[SUS-11] The suspension type is “b – Partially suspended penalty/measure” or “d – Partially suspended penalty/measure with probation/supervision” while the suspension duration is equal or superior to the duration of execution of the sanction.

[SUS-12] The suspension type is “a – Suspended penalty/measure” or “c - Suspended Penalty/measure with probation/supervision” while the suspension duration is inferior to the duration of execution of the sanction.

Example:

Sanction "1001 –Deprivation of freedom" is 14 months with suspension where the suspension duration is for 8 months while the type of the suspension is not indicated as "partial".

[SUS-13] The end date is earlier than (start date + duration)

[SUS-14] The suspension probation duration has a value > 0 while suspension type indicates "a – Suspended penalty/measure" or "b – Partially suspended penalty/measure".

[SUS-15] The suspended amount of the fine indicates a value > 0 while the number of fines indicated in the related sanction is superior to 1.

[SUS-16] The suspended amount of the fine indicates a value > 0 while the category of the related sanction does not reference the category "8000 - Financial Penalty category code" or one of its child categories.

[SUS-17] The suspended amount of the fine is equal or exceeds the amount of individual fine indicated in the related sanction.

[SUS-18] The suspended amount of the fine indicates a value > 0 while the suspension type is not "b – Partially suspended penalty/measure" or "d – Partially suspended penalty/measure with probation/supervision" (i.e. the suspension is not declared as partial).

"Interruption" entity

[INT-1] The end date is before the start date.

[INT-2] The start date is before the decision date of the conviction.

[INT-3] The start date is before the final decision date of the conviction.

[INT-4] The start date is before the execution start date of the sanction.

[INT-5] The end date is before the decision date of the conviction.

[INT-6] The end date is before the final decision date of the conviction.

[INT-7] The end date is before the execution start date of the sanction.

[INT-8] The end date is earlier than (interruption start date + duration).

Requests

[REQ-1] Request purpose category = "001 – criminal proceedings" while the requesting authority type = "A – competent administrative authority", "P – person concerned for information on own criminal records" or "E – Employer"

[REQ-2] The new deadline provided in the "request deadline" message does **not** fulfil the following conditions:

- For proceedings requiring a maximum of 10 working days (the requesting authority type is "J – judicial authority" or "A – competent administrative authority"):
  - deadline < (reception of request message + 16 calendar days)
  - deadline < (reception of additional ID info message + 16 calendar days)

  (Contingency used for future date is 1 week-end + 4 office closing days)
- For proceedings requiring a maximum of 20 working days (the requesting authority type is "P – person concerned for information on own criminal records"):
  - deadline < (reception of request message + 34 calendar days)

  (Contingency used for future date is 3 week-ends + 8 office closing days)

Common Reference Tables

[CRT-1] One of the following information elements transmitted does not contain a value from the corresponding common reference table:

- Place: Country → common reference table "Countries and Nationalities"
- Place: Town Code → common reference table "Cities"
- Place: Country Subdivision → common reference table "Country Subdivisions"
- Person: Nationality → common reference table "Countries and Nationalities"
- Sanction: Currency of Fine → common reference table "Currencies"
- Request message: Purpose Category → common reference table "Request Purposes"

- Requesting Authority: Type → common reference table "Requesting Authorities"
- Identification Document: Identification Document Category → common reference table "Identification Document Categories"
- Conviction: Convicting Country → common reference table "Countries and Nationalities"
- Offence: Common Category → common reference table "Offences"
- Offence: Level of Completion → common reference table "Levels of Completion"
- Offence: Level of Participation → common reference table "Levels of Participation"
- Sanction: Common Category → common reference table "Sanctions"
- Sanction: Type → common reference table "Types of Sanction"
- Sanction: Alternative → common reference table "Types of Alternative Sanctions"
- Suspension: Type → common reference table "Types of Suspension"
- Decision Change: Type → common reference table "Types of Decision Changes"
- Request Response message: Other Member State → common reference table "Central Authorities"

[CRT-2] Usage of expired reference values:

- For the "Currencies" common reference table: the "Valid To" information field of the currency value used in currency of fine is before the decision date of the conviction.

  Example: A person of Belgian nationality, convicted on 31.08.2014 in Slovakia, is the subject of a notification transmitted by Slovakia to Belgium on 4.10.2014. One of the sanctions transmitted is the payment of a fine of 5000 Slovak koruna. As it can be easily observed, from the Common reference table spread-sheet, this is a functional error as Slovak koruna was valid only up until 01.01.2009.

- For the "Countries" common reference table: the "Valid To" information field of the country value used in the birth place is before the birth date.

  Example: A person, subject of a notification in 2014, is born on 10.02.1993 and has as country of birth the value "Yugoslavia". This represents a functional error, since the value Yugoslavia was valid until 01.01.**1992**.

  Please note that for countries, an entry may be at the same time valid as country of birth of the person referred in the message but not as current nationality. As an example, the value "Czechoslovakia" can be a valid country of birth in a message if the person was born before 1993 but cannot be used in the same message as being the current nationality.

  The same rule is not to be applied to the "Cities" and "Country Subdivisions" common reference tables. Indeed, for these fields such a rule may be counterproductive. If for example a message uses the current name of a city as town of birth whereas the city was named differently on the date of birth of the person, it can still prove useful to keep this information so as to find the person in the national registers.

Logging, Monitoring and Statistics

[LMS-1] An ECRIS message is received for which the timestamp indicates that it was sent during the interval after time 21:59:59 of the last calendar day of the month and before time 02:00:01 of the first calendar day of the month.

According to the "Logging, Monitoring and Statistics Analysis" document, this specific condition must raise a specific functional error code.

5.5.2 « Manual » Validation

The evaluation of the following conditions cannot be automated in the ECRIS software. However when encountering such a situation where the ECRIS message obviously contains an inconsistency, the end-user should be able to reject the faulty message and to respond to the sender with the most appropriate functional error code.

[MAN-1] The information received in a message successfully passes the automated technical and functional validations but the data contained in the information elements is unusable.

A few examples:

- Offence National Title = "GRTSDSRDAH" and Offence Applicable Legal Provisions = "jckzekfhfkfehkfe" and Offence Start Date = "dummy"
- Decision Date = "dummy", Final Decision = "dummy", Convicting Authority Name = "QWERTYU", Convicting Authority Code = "00000", File Number = "abcde/abc"
- A request message containing only dummy values for the mandatory elements
- Person forename = "Driving" and person surname = "Licence"

[MAN-2] The national title of the offence does not match the common category that has been provided.

Example:
Common category = "0301 00 – Directing a terrorist group" and national title = "Polygamy"

[MAN-3] The national title of the sanction does not match the common category that has been provided.

Example:
Common Category = "1001 – Imprisonment" and National Title = "Fine"

[MAN-4] The additional identification information provided does not correspond to the initial identification information provided in the notification or request message.

[MAN-5] A notification message providing an update of a conviction previously notified correlates with a previous notification message but the new conviction information contained does not match at all with the information of the conviction that is supposed to be updated.

[MAN-6] The cancellation of the related message cannot be handled.

Please note that **no functional error** is foreseen for the case where a notification message conveying an update to a conviction does **not** correlate to a previously notified conviction (i.e. meaning that the technical identifier indicated in the property "MessageDataUpdatedConviction" does not match the technical identifier of any previously notified conviction – identified either through the unique ECRIS identifier of the conviction, the unique NJR identifier of the conviction or a structure formed of the code and name of the convicting authority, the file number of the conviction and the final date of the conviction). Indeed, following the snapshot approach already defined in the "Business Analysis" document, a notification message is always complete and contains all up-to-date information about the conviction. In this specific case where the notification message indicates that it provides an update but that the initial conviction is not known by the receiver, the notified Member State should simply treat the message as if it was the notification of a new conviction.

5.6 Guidelines and Recommendations

5.6.1 Implementation Platforms

As mentioned earlier, the technical artefacts of the detailed technical specifications have been tested using various development platforms. These platforms have been chosen based on their maturity, market penetration and preferences indicated by the Member States in the "Inception Phase Questionnaire".

It is necessary to underline that the technical artefacts that constitute the ECRIS detailed technical specifications (i.e. the WSDL and XSD files) are **fully compliant** with the specifications established in the "Technical Architecture" document and are **purposefully not designed for accommodating specific development platforms**. The aim of the technical artefacts is to stick at 100% to the specifications defined in the "Technical Architecture" document so as to remain neutral towards the development platforms.

Tested platforms

Concretely, the following table provides an overview of the platforms used with their according versions:

Platform	Platform / Vendor URL	Version
Apache AXIS1	http://ws.apache.org/axis/	1.4
Apache AXIS2	http://ws.apache.org/axis2/	1.5.2
Apache CXF	http://cxf.apache.org/	2.3.0
Spring WS	http://static.springsource.org/spring-ws/sites/1.5/	1.5.9
JAX-WS RI (Metro Core)	https://jax-ws.dev.java.net/	2.2.1
.NET WCF	http://msdn.microsoft.com/en-us/netframework/aa663324.aspx	4.0

Table 4 – Implementation platforms and their versions

The operating system used for the tests is Microsoft Windows version 6.1.7600.

The *JDK* that has been used so as to program and compile the Java classes produced by the Java-related platforms is version 1.6.0_13. In addition, the Apache Maven software project management and comprehension tool version 3.0 has been used to handle automated builds and to support basic testing.

For the development and testing of the .NET WCF platform, Visual Studio 2010 version 10.0.30319 has been used. The artefacts for .NET were generated using "svcutil" version 3.0.4506. In addition, the plug-in WCF Blue (http://wscfblue.codeplex.com/) for Visual Studio was also used to generate the artefacts within Visual Studio.

Main conclusions

Overall, the tests performed allowed to proactively fix possible incompatibilities of the tested platforms with the technical artefacts. It must be noted here that the term "incompatibilities" does not refer to the technical artefacts but solely to the development platforms. Indeed, each development platform is a software product implementing the *web services* specifications. Thus, the quality of the implementation in regards to the specifications provided by *W3C* for *web services* depends solely on how the developers that were involved in that task perceive the said implementations. Please note also that these incompatibilities spotted during practical testing are not blocking for properly producing and consuming the ECRIS messages but may require specific minor adaptations to the development platform itself in order to by-pass these minor issues.

Beyond testing for incompatibilities, during this process the platforms were also evaluated for their extensibility and compatibility. Based on that evaluation, the following recommendations can be made:

- It is strongly recommended that implementers update their implementation platforms up to the latest available and stable version.

  Using the latest stable version of a given development platform will ensure that common issues and well-known bugs have been fixed and thus this platform will implement more closely the technical specifications to be used.

- For the JAVA related frameworks, it is recommended that implementers consider Spring WS and Apache CXF for their implementations.

  Indeed AXIS1 provided its latest delivery on 22 April 2006 and can safely be considered a deprecated project. AXIS2 is a mature and stable framework but the releases provided up to now are mainly maintenance releases with very long release cycles. Version 1.5.1 was released on 23 October 2009 whereas the latest stable version 1.5.2 was released only on 06 September 2010. The JAX-WS RI provided by Oracle provides only basic functionality and the integration and extensibility options that are supported are minimal.

  Apache CXF and Spring WS on the other hand are actively maintained, integrate easily with other technologies and are easy to extend without requiring substantial effort. Additionally both frameworks are open source, with support from large communities and the available knowledge base on Internet for is quite significant.

Regardless of the recommendations made above, please note that all JAVA platforms that implement properly the specifications, standards and protocols defined in the "Technical Architecture" document will not have problems in consuming or producing messages based on the technical artefacts of ECRIS.

- If automatic tools are used so as to generate code from the technical artefacts, it is also recommended that implementers perform thorough reviews of the produced code so as to ensure their validity.

  This recommendation is based on the fact that during the testing, some tools have been found to interpret and generate the specifications in an erroneous way. This is the case mostly for the Axis platforms, especially Axis1 which faced difficulties in producing automatically code that would compile. Axis2 on the other hand, on version 1.5.1 when using "adb" for serialisation, could not produce classes for type definitions that were deriving by restriction from their parent types. This issue could be avoided by moving to a newer version of Axis2.

In addition, attention should be paid to the service interfaces that are generated by the automated tools, especially due to the fact that the operations defined in the *service contract* are "one-way" operations but can potentially return faults. It has been noted that some tools like Apache WCF or JAX-WS RI flag such operations using annotations or attributes which explicitly define them as being "one-way". In doing so, the frameworks tend to "swallow" faults returned by the application code rather than returning them to the caller.

Impacts on *service contract*

Lastly, it must be noted that specific decisions on some details of the design of the *service contract* have been taken so as to mitigate problems that some development platforms encounter when processing it.

More concretely, the optional "name" attribute of the element "<wsdl:input>" must be used and must contain identical values, both under the "portType" and "binding" definitions of the document. This was necessary so as to avoid a well-known problem that "svcutil" from the .NET framework encounters when processing *WSDL* files which define a different name in the "portType" "<wsdl:input>" element than the one used in the "binding" (for more information on that subject, please refer to http://social.msdn.microsoft.com/Forums/en-US/wcf/thread/626bde79-45c4-4ae2-ba7d-dd7834bacd60).

Also, even though the operation "isAlive" should follow the "notification" primitive defined in the *WSDL* specification 1.1 (see also http://www.w3.org/TR/wsdl#_notification) it was not possible to define the service using the appropriate grammar because Apache CXF would not support it. For that reason, the operation has been defined as requiring an input message. This input message can however be ignored by the ECRIS software implementations.

5.6.2 "isAlive" Operation

As defined in the "Technical Architecture" and repeated in this document, the "isAlive" operation allows a Member State to check whether another Member State's end-point can be reached and whether it is properly responding. Implementers of this operation can essentially use it as a flag signaling whether their ECRIS system can accept and process appropriately messages.

Even though no strict policies are applied on the *service contract* or defined in this document in regards to the invocation of this operation, implementers should take under consideration the following suggestions:

- The "isAlive" service should be invoked within regular intervals so as to monitor the availability of other ECRIS implementations. The regular intervals should range between 6 and 12 hours so as to allow taking corrective measures in case an ECRIS instance is found to be unavailable.
- It is likely that several Member States will first pile up a series of notifications before sending them all out during a predetermined moment later in time. This results thus in a

significant amount of messages being sent during a specific and predetermined period of time. It is thus recommended to the implementers to use the "isAlive" operation **once before** initiating such a **grouped sending of messages**. Indeed, in the case that the receiving software is unavailable, following this recommendation will greatly reduce the number of failed messages exchanged as well as reduce the resource usage of the sending Member State. Similarly, the "isAlive" operation should not be systematically invoked before sending each business message so as to avoid multiplying unnecessarily the number of messages.

- The implementation of the "isAlive" service should not only indicate whether the web service end-point alone is up and running. It should also check the state of the back-end systems participating in the processing of the ECRIS messages and should return a negative message if the chain of systems cannot completely process the messages.

A positive side effect of using the "isAlive" service in the ways suggested above is the fact that other errors, not directly related with the availability of the requested ECRIS instance, can be spotted and can allow taking corrective measures proactively. For example, the "isAlive" service must reply with a "soap:fault" using the "WrongVersionUsed" code in case the calling ECRIS end-point is using an older version of the service than the one it should use. If such a fault is detected, the calling ECRIS instance should immediately log the incident and the system operators should be notified so as to rectify the problem.

5.6.3 Retry Policies

It is a common practice in message exchanges to retry a message transmission if it has failed in a previous call, especially when the cause of the failure related to the network connectivity. Such retries are not explicitly defined by the detailed technical specifications, since such policies are specific to the implementations and are not declared in the *service contract*. In particular, it is **not** required for an ECRIS instance to necessarily support such policies.

The reasons a message exchange could fail in ECRIS have to be viewed both from a technical and from a business perspective. The difference between the two perspectives is that the business retransmissions are usually foreseen as part of the process, if for example a business error has occurred. On the other hand, the technical perspective usually foresees automatic retransmission of messages in case of technical failures. These cases include general availability problems (i.e. connectivity, ECRIS instance replied with "false" during an "isAlive" operation, etc.) as well as technical errors provoked by the receiver, such as considering a message structurally invalid whereas the sender has already structurally validated the message or the HTTPS certificate provided by the server was found invalid.

On the technical level, only **availability issues** should lead to **automated retransmission** of failed messages. Indeed, technical problems provoked from the receiver, such as considering a structurally valid message as invalid, usually requires human intervention in order to be rectified.

In addition, it is also advisable that implementers follow common practices for the automated retransmission policies. It is strongly suggested **not to immediately retry** sending a failed message, but rather to keep a queue or back-log of failed messages and retry at a later moment in time. It is also advised to increase the time interval with each attempt. Concretely, if a first attempt to resend a failed message occurs 1 hour after the initial transmission, it is recommended that the next attempt is made only after 3 hours. Following this practice should again reduce the number of failed messages. Also, the time interval between retries should be sufficiently large so as to allow system operators to rectify the problems that blocked the initial message transmission.

5.6.4 Traffic Management

Please note that at the time of writing of this document, no reliable figures are available on the expected network traffic or number of messages to be exchanged after the go-live of the ECRIS implementations in April 2012.

Obviously, the number of messages exchanged varies greatly from Member State to Member State due to specific factors such as the size of the country, the number of borders with other

Member States, the movements of people of various nationalities, etc. However, from the answers received to the "Inception Phase Questionnaire" and from the intermediate NJR statistics of 2010, it can be observed that the **current** maximum number of notifications and requests exchanged ranges up to approximately 96.000 per year for one Member State. Considering the fact that in ECRIS, at least 2 *web service* operations are required for completing properly a notification transaction and at least 3 are required for completing a request transaction, the current volume of exchanges would already imply that that Member State would need to process at least approx. 16.000 *web service* operations per month.

The figures above are however only representing a very rough extrapolation of the current situation. The fact that a significant increase of message exchanges is expected by most Member States once ECRIS is operational also needs to be taken into account. This expected increase is mainly due to the facts that all EU Member States will be participating in the exchanges and that ECRIS provides support for handling requests for non-criminal proceedings. In addition to this, the currently on-going project ECRIS-TCN (Third Country Nationals) is also exploring the possibility to leverage in the future the available functionality of ECRIS for realising additional criminal records information exchanges on non-EU citizens.

Based on the figures above, and taking a large buffer margin, it can be reasonably assumed that the number of *web service* operations could range up to 50.000 per month for a given Member State.

Please note that during the tests performed when designing and implementing the detailed technical specifications, the approximate size observed for an XML message without binary attachment is of 6Kb. As a reminder, the maximum size has been limited to 10 Mb in the "Technical Architecture" document.

Given the figures above, solid and mature frameworks and systems should not be easily overwhelmed by the expected amount and volume of messages. However, difficulties may occur when high peaks of messages are sent and/or received.

Thus, in order to mitigate these risks, the following recommendations can be made:

- On the receiver's end, it is recommended to establish priorities for the processing of incoming messages. Generally, messages related to the "request" process should be treated with higher priority than messages related to "notifications". Within the group of "request"-related messages, it is also recommended to put the highest priority on the processing of the "Request" and "Request Deadline" messages.
- On the sender's end, it is also strongly recommended that ECRIS implementations do **not** send all the volume of their messages at the same time of the day but rather prioritise the messages, following the same priorities as defined above, and send them in smaller groups at regular intervals.
- In order to establish more accurate figures in the future for performing traffic management, implementers are advised to monitor and establish statistics on the network bandwidth used, sizes of the files being exchanges and number of messages per type.

6 FUTURE EVOLUTIONS

This chapter briefly provides an overview of the approach that to be followed when creating a new version of the technical artefacts.

Please note that a detailed description on the versioning strategy and its implementation in the context of ECRIS has already been provided in the "Technical Architecture" document. This chapter does not aim to alter in any way the approach already agreed upon but rather to make more explicit how it should be performed in practical terms.

6.1 Versioning of WSDL/XSD Files

Each of the technical artefacts already implements the versioning concepts. All files have been suffixed with the version identifier "v1.0". The namespaces defined in the *XSD* and the *WSDL* files are also suffixed with the same version identifier. The definitions of the operations and messages provided in the *WSDL* are also suffixed with the same version identifier.

Assuming that a minor version of these technical specifications needs to be produced, the following actions should be performed by the central entity:

- Implement the changes in the *XSD* and *WSDL* files and suffix these files with the incremented version identifier.
- Suffix all the namespace declarations in the *XSD* and *WSDL* files using the new version number.
- Create a copy of the messages, bindings, operations and service declarations within the *WSDL* document and suffix them accordingly using the new version identifier (the previous declarations with the old version must remain in order to support two concurrent versions).
- If necessary, remove the oldest definitions of messages, bindings, operations and service declarations from the *WSDL* document (only if the WSDL would otherwise support more than two concurrent versions).
- Document and validate the changes performed in the technical artefacts.
- Provide the new versions of the technical artefacts to the Member States' experts for revision.

Once the new version of the technical artefacts is agreed upon, and once a Member State has implemented this new version in its national ECRIS software, the Member State must modify the newest version of the *WSDL* file and update the "location" attribute of the "soap12:address" for each version of the "wsdl:service" element so as to properly reflect the *HTTP* address that other Member States should use in order to send ECRIS messages. These updated service contracts must be sent to the central entity, along with any other information that can affect the other Member State (such as the first date when the new implementation will be deployed or whether there was a change in the implementation of binary attachments).

As a final step, the central entity validates the newly received *WSDL* file and disseminates it to all other Member States. During this validation, the central entity should verify at least the following points:

- that the WSDL file version and version information used in the namespaces are still correct
- that the WSDL file has not been altered in the definition of messages and operations
- that the end-point declared in the WSDL file indeed corresponds to an end-point that is accessible through the sTESTA network

6.2 Changing Common Reference Tables

Please note that the rules to be strictly applied when changing the content of common reference tables have already been detailed in the "Technical Architecture" document.

In addition, the technical approach to be followed when modifying the common reference tables is similar to the one described earlier for producing new versions of the other technical artefacts. Changing either **the structure or the content** of the common reference tables **must always** result in a new minor version. This approach is preferred because it does not require additional kinematics for establishing which version of the common reference tables is supported by a specific ECRIS software implementation.

7 ANNEX – OVERVIEW OF MESSAGE VOLUMES

7.1 Overview of Member States' Answers

- The following provides an overview on the answers provided by the Member States' central authorities to the specific question of the "Inception Phase Questionnaire" on the expected volume of information that would need to be exchanged, assuming that ECRIS would be used currently:
  - Approx. 96.000 notifications and requests per year: PL
  - Approx. 55.000 notifications and requests per year: FR
  - Between 25.000 and 50.000 notifications and requests per year: ES, UK
  - Between 10.000 and 25.000 notifications and requests per year: BE, CZ, LT, LU, SE
  - < 10.000 notifications and requests per year: EE, FI, HU, RO, SI, SK
  - Unknown/not provided: AT, DE, NL, PT
- Do you foresee an increase of exchanges of criminal record data with other Member States once the ECRIS software is operational in all Member States?
  - Yes: AT, BE, CZ, ES, FI, FR, HU, LT, LU, NL, PL, PT, RO, SE, SK, UK
  - No: DE, SI
  - Unknown: EE

7.2 Overview of NJR Statistics 2010

The following figures have been extracted from the consolidated statistics of the NJR message exchanges for the period ranging from January until July 2010, thus 6 months of activity.

Member State	Notifications	Requests	Total
BE	5862	2457	8319
BG	2131	1444	3575
CZ	4708	2943	7651
DE	22089	18699	40788
ES	5652	1736	7388
FR	12134	7998	20132
IT	1944	228	2172
LU	3313	1712	5025
PL	14468	9988	24456
SK	3037	1877	4914

Please note that this table indicates for notifications and requests both the messages **sent and received** by the NJR Member State during the defined period of time. The column providing the total values indicates simply the sum of notification and request messages sent and received by each Member State during the given period of time.

This implies in particular that a given NJR message is counted twice in this table: once for the Member State that has sent it and once for the Member State that has received it. This is correct because this table mainly aims at showing the order of magnitude of the number of messages to be treated for each Member State, rather than providing exact counts of the NJR messages.

____