@GIT Initiative

“Standardizing Telemedicine”

Recommendation

for a Standardized

Teleradiology Transmission Format

Version 1.7

www.tele-x-standard.de
teleradiologie@drg.de

1 Table of Contents

2 Table of Figures

3 Imprint

Initiative “Standardizing Telemedicine”
Working Group on Information Technology of the German Radiology Society
(Arbeitsgemeinschaft Informationstechnologie der Deutschen Röntgengesellschaft)
E-mail: teleradiologie@drg.de
URL: http://www.tele-x-standard.de

Title: Recommendation for a Standardized Teleradiology Transmission Format
Version: 1.7

3.1 Active members of the working group (in alphabetical order)

Name	Organization	Versions
Avtos, Gatis	Steinhart Medizinsysteme	1.7
Baur, Stefan	Curagita AG, Heidelberg	1.1,1.5
Engelmann, Uwe	CHILI GmbH, Heidelberg	1.1,1.5,1.6,1.7
Erdmann, Stephan	abasoft EDV-Programme GmbH, Korntal	1.7
Gamroth, Christian	abasoft EDV-Programme GmbH, Korntal	1.7
Kämmerer, Marc	VISUS GmbH, Bochum	1.1,1.5,1.6,1.7
Klos, Gordon	Klinik und Poliklinik für Diagnostische und Interventionelle Radiologie, Universitätsmedizin Mainz	1.1,1.5,1.6,1.7
Köster, Claus	GI Gesundheitsinformatik GmbH	1.1,1.5,1.6
Kreisel, Roman	Abasoft EDV-Programme GmbH	1.6
Mildenberger, Peter	Klinik und Poliklinik für Diagnostische und Interventionelle Radiologie, Universitätsmedizin Mainz	1.1,1.5,1.6,1.7
Münch, Heiko	CHILI GmbH, Heidelberg	1.1,1.5
Lehr, Martin	Abasoft EDV-Programme GmbH	1.7
Pelikan, Ernst	Universitätsklinikum Freiburg, Klinikrechenzentrum	1.1,1.5
Philipps, Mario	Steinhart Medizinsysteme GmbH	1.1,1.5,1.6
Ruggiero, Stephan	Institut für Klinische Radiologie und Nuklearmedizin Universitätsmedizin Mannheim	1.1,1.5
Runa, Alain	Institut für Klinische Radiologie und Nuklearmedizin Universitätsmedizin Mannheim	1.1,1.5
Schäfer, Marc	Institut für Klinische Radiologie und Nuklearmedizin Universitätsmedizin Mannheim	1.7
Schneeberg, Sven	VISUS GmbH, Bochum	1.6,1.7
Schröder, Stephan	CHILI GmbH, Heidelberg	1.1,1.5
Schröter, Andre	CHILI GmbH, Heidelberg	1.1,1.5,1.7
Schütze, Bernd	http://www.medizin-informatik.org/	1.1,1.5,1.6
Schwind, Florian	CHILI GmbH, Heidelberg	1.6,1.7

Walz, Michael	Ärztliche Stelle für Qualitätssicherung in der Radiologie Hessen	1.1,1.5,1.6,1.7
Weisser, Gerald	Institut für Klinische Radiologie und Nuklearmedizin Universitätsmedizin Mannheim	1.1,1.5,1.6,1.7
Westermann, Michael	GI Gesundheitsinformatik GmbH	1.1,1.5,1.6

Last revised: 31.07.2013

4 Contact

The working group can be contacted at any time via the following e-mail address: *teleradiologie@drg.de*. Interested parties will be added to our mailing list upon request.

Personal data is kept strictly confidential and never will be passed on to a third party.

Results achieved by the Initiative will be published on the website at *http://www.tele-x-standard.de*.

5 Copyright

6 Preamble

Members of the @GIT Initiative “Standardizing Telemedicine” have come together with the intention of developing a recommendation for a communication protocol suitable for teleradiology. The members come from various academic and research institutions as well as from the industry. Anyone who wants to contribute can join the group.

Results are fully accessible and can be used by the public now and in the future (Public Domain).

7 Preface – Version 1.7

This is the first version of this White Paper which will be issued in English only.

This White Paper introduces the use of the DICOM E-MAIL for building and maintaining telemedicine networks that can be administrated centrally. The key element to achieve this goal has been the extension of the pre-existing Service Parts. Consequently most of the work went into the definition of the new Service Parts:

- ADDRESSUPDATE – CLEAN, GET
- CONTACTUPDATE – CLEAN, GET,REMOVE, SET
- CONNECTIONUPDATE – GET
- DISPOSITIONNOTIFICATION
- KEYUPDATE – CLEAR, GET
- QUERY – FIND, RESULT

Part of the extension of the Service Parts has been the introduction of a new notification mechanism using Service Part e-mails as notification e-mails replacing the previous existing notification mechanism type 2. In contrast to its predecessor, the use of the notification mechanism type 3 is mandatory in order to be compliant with this version 1.7 of the Recommendation for a Standardized Teleradiology Transmission Format.

A minor change has been introduced regarding the use of OpenPGP key IDs. Unless otherwise marked, the key ID now refers to the long version, which is represented by the last 16 digits of the 40 digit fingerprint.

The authors of the Working Group,

July 2013

8 Preface – Version 1.6.1

This is the first English version of the Recommendation for a Standardized Teleradiology Transmission Format. In comparison to version 1.6, this version has been translated from German and errors have been corrected (cf. 11 Errata).

The authors of the Working Group,

March 2013

9 Preface – Version 1.6

The present version 1.6 of the recommendation deals with the support of administrative tasks and quality testing.

A German standard for quality assurance in Teleradiology (DIN 6868-159) was published in March 2009. Since then, according to the Röntgenverordnung (RöV, German X-Ray Ordinance), it is required to perform quality and constancy tests when sending data in Teleradiology networks. Particularly in heterogeneous networks using software from different manufacturers, these requirements represent a new challenge to perform the requested tests automatically. The basic task here is the exchange and processing of communication data across multiple vendors.

A second challenge is the convergence of existing networks. It becomes necessary to not only exchange DICOM images but also administer messages that are sent between partners to enable the administration of growing networks.

In order to meet these new requirements, so-called *Service Part e-mails* are added in version 1.6. Based on the previous standard recommendations introduced in versions 1.1 and 1.5, the following scenarios can be presently covered:

1. Vendor independent communication of the required data for quality and constancy tests according to DIN 6868-159.
2. Exchange of PGP/GnuPG key data across vendors.
3. Exchange of address data across vendors.

For completed actions, confirmations are sent in accordance with the notification mechanisms described in version 1.5.

The authors of the Working Group,

June 2010

10 Preface – Version 1.5

Version 1.1 has already enabled the transfer of DICOM and NON-DICOM data across different vendors using the e-mail protocol. The present version 1.5 covers the following additional extensions:

1. A mechanism verifying the reception and completeness of data on the side of the receiver. For this purpose, additional information is added by the sender to the data packets, which can be analyzed, upon receipt.
2. A mechanism verifying the reception and completeness of data on the side of the sender. For this purpose, confirmation notifications including a defined status, are sent back to the sender by the receiver.
3. A mechanism verifying the authenticity of the sender and the integrity of the data packets. This allows identification of modifications on data packets made by third parties, and refusal of acceptance for unwanted data packets.

The authors of the Working Group,
October 2005

11 Errata

11.1 Version 1.6 / 1.6.1

11.1.1 Error in Figure – Example of a Service Part trigger e-mail

Fig. 65 – Example Service Part trigger e-mail contained an error. The value of X-TELEMEDICINE-SERVICEPART should read TESTRANSFER instead of REQUEST-FOR-TESTTRANSFER. The following Fig. 1 – Incorrect value of X-TELEMEDICINE-SERVICEPART and Fig. 2 – Corrected value of X-TELEMEDICINE-SERVICEPART show the incorrect and the corrected value.

```
X-TELEMEDICINE-SERVICEPART: REQUEST-FOR-TESTTRANSFER
```

Fig. 1 – Incorrect value of X-TELEMEDICINE-SERVICEPART

```
X-TELEMEDICINE-SERVICEPART: TESTTRANSFER
```

Fig. 2 – Corrected value of X-TELEMEDICINE-SERVICEPART

11.1.2 Error in Figure – Sample process: Constancy test by Service Part trigger e-mail

The flow chart was incorrect as the notification e-mail as a response to the reception of the trigger e-mail will be sent upon reception and is not connected to the test data transfer status. The flow chart has been altered (Fig. 64 – Sample process: constancy test by Service Part trigger e-mail).

11.2 Version 1.5

11.2.1 Error in Figure – MDM according to RFC 3798

The following example contains an error in the machine-readable paragraph in which header and body are delimited (cf. arrow in Fig. 3 – Incorrect confirmation of receipt according to RFC 3798). A delimiter needs to be introduced here by adding a new line after the Content-Type. This error needs to be corrected in the software starting with the adoption of the standards recommendation version 1.6 (in accordance with the example Fig. 9 – Confirmation of Receipt compliant with RFC 3798).

```
Date: Mon, 1 Jan 2009 00:19:00 (EDT) -0400
From: radiology_mannheim@teleradiology.de
Message-Id: <2323423432019.12345@teleradiology.de>
Subject: Disposition notification
To: radiology_mainz@teleradiologie.de
MIME-Version: 1.0
Content-Type: multipart/report; report-type=disposition-notification;
boundary="RAA14128.773615765/teleradiology.de"
```

```
--RAA14128.773615765/teleradiology.de

 Here is the spot for additional optional text

--RAA14128.773615765/teleradiology.de
content-type: message/disposition-notification
Reporting-UA: post.teleradiology.de; Mailprogram 1.1
Final-Recipient: rfc822; radiology_mainz@teleradiologie.de
Original-Message-ID: <199509192301.23456@teleradiologie.de>
Disposition:automatic-action/MDN-sent-automatically;displayed/warning
Warning: 1.2

--RAA14128.773615765/teleradiology.de--
```

Fig. 3 – Incorrect confirmation of receipt according to RFC 3798

11.2.2 Request Receipt Mechanism 2 - X-TELEMEDICINE confirmation (optional)

Please note that this chapter has been rewritten and is now referenced as 17.4.2.2 E-mail notification mechanism 2 (obsolete).

X-TELEMEDICINE-DISPOSITION-NOTIFCATION-TO should read *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO.* The same applies to *X-TELEMEDICINE-DISPOSITION-NOTIFCATION-KEYID* which should read *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-KEYID.*

As of version 1.6.1, both spellings of the X-tags must be supported. From version 1.8 onwards, only the corrected spelling will be valid.

11.2.3 Errata in table 20. Appendix Error Codes

1.3	mail-attachement-error -> mail-attachment-error
1.3.1	mail-attachment-corrupt -> mail-attachment-corrupt
3.2.1	application-intern-attachement-error ->application-intern-attachment-error
3.2.1.1	application-intern-attachement-not-processed -> application-intern-attachment-not-processed

12 Basic consideration

The recommendation is based on the assumption that the exchange of e-mails with encrypted data provides the smallest common denominator for a secure exchange of non-anonymized or non-pseudonymized data.

13 Minimum requirements for the software

13.1 Data type transfer

In general, it must be possible to transmit any type of data (DICOM and NON-DICOM). If the receiving software cannot handle a particular type of data, it should be forwarded to another application or stored temporarily. In any case, receipt of unknown objects must not lead to an abortion of the data transmission.

13.2 MIME Standard

The software must support the MIME Standard[1], in particular Multipart Mail, Message Partial, as well as use of X-tags.

14 Extended requirements for the software

The secure variants of the e-mail protocols (SMTP, POP3 or IMAP4) should be supported.

15 Encryption & compression

Encryption of data must be compatible to OpenPGP (RFC 4880). The use of compression as defined in RFC 4880 is optional, but recommended. A guideline for the encryption key length and algorithm, which is considered as secure, has been published by the German Federal Office for IT Security[2].

16 Digital signature (PGP/MIME)

PGP/GnuPG can be used for encryption as well as for signing of data. In the context of this recommendation, the following two methods are recommended in accordance with RFC 3156 and RFC 1847:

[1] In particular RFC2045/46 (MIME Part 1&2), RFC3156 (MIME Security with OpenPGP); Source: http://www.ietf.org/

[2] https://www.bsi.bund.de/ContentBSI/Publikationen/TechnischeRichtlinien/tr02102/index_htm.html

- E-mail encryption combined with signature (RFC 3156, chapter 6.2 – *Combined method*)
- Data with a separate signature, which is encrypted compatible to the OpenPGP standard in accordance with RFC 1847. (RFC 3156, chapter 6.1 – *Encapsulation*)

In accordance with this recommendation all other e-mails must be signed, except for MIME conform notify e-mails.

17 Transfer format

17.1 DICOM data

DICOM data is transformed into a DICOM MIME type (application/dicom) and attached to an e-mail in accordance with Suppl. 54 of the DICOM Standard and must be multipart/mixed. The resulting e-mail is encrypted and signed compatible to the OpenPGP standard (Fig. 4) (cf. Chapter 16 – *Digital signature (PGP/MIME)*).

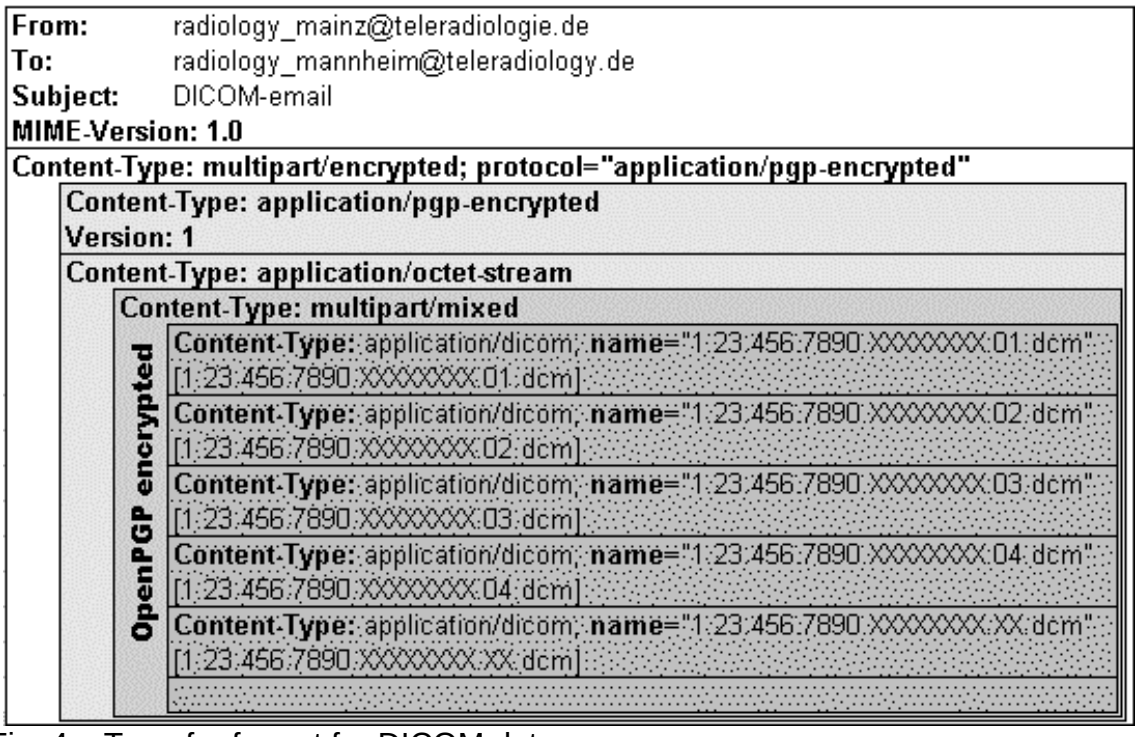

Fig. 4 – Transfer format for DICOM data

17.2 NON-DICOM data

Used X-tags: X-TELEMEDICINE-STUDYID

NON-DICOM data does not contain a header with patient information. For this reason, a new tag *X-TELEMEDICINE-STUDYID* has been defined, conforming to

the MIME Standard. This tag is used to transmit the StudyInstanceUID in order to assign a NON-DICOM object to a DICOM study. In case there is no suitable DICOM study or StudyInstanceUID, a new UID needs to be generated and assigned to the NON-DICOM data (cf. chapter 20 Generating identification numbers). All NON-DICOM data is encrypted and signed in compliance with the OpenPGP standard together with all DICOM data (Fig. 5 – Transfer format for NON-DICOM data). This facilitates the shared transmission of multiple exams or multiple patients with a single e-mail. It is also possible to send associated data in different e-mails while the matching is done via the respective StudyInstanceUID (in case of DICOM data) and *X-TELEMEDICINE-STUDYID* (in case of NON-DICOM data).

Use of the *X-TELEMEDICINE-STUDYID* for DICOM data is not permitted and must be ignored on the part of the receiver (Rule of unambiguousness).

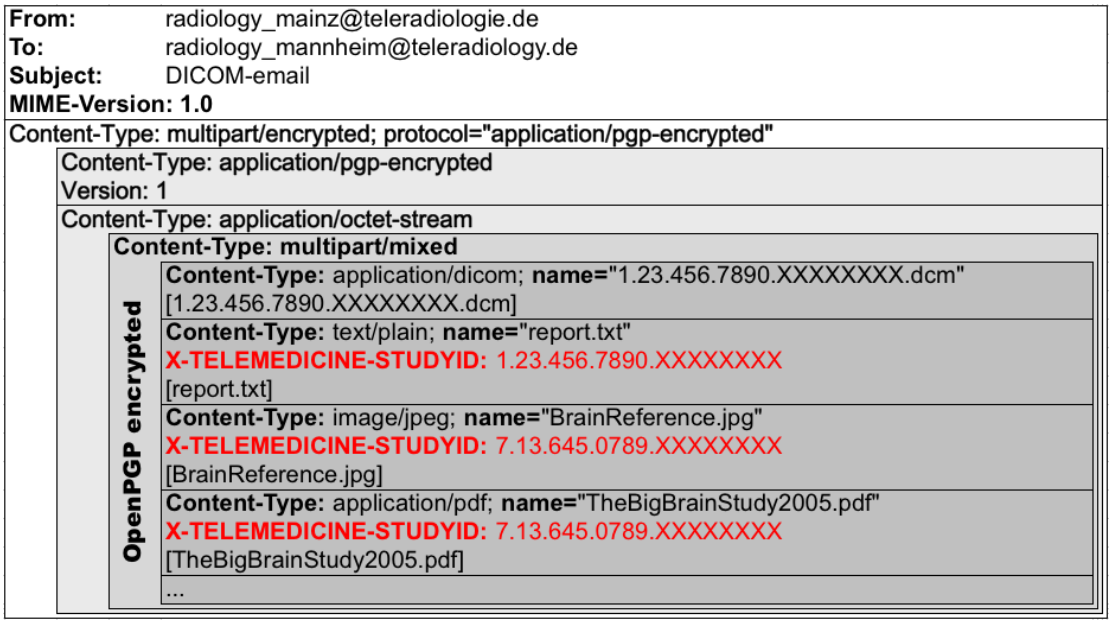

Fig. 5 – Transfer format for NON-DICOM data

17.3 Message/Partial

In the overview (Fig. 6 –) data is sent encrypted as a PGP/MIME Multipart e-mail. The MIME Standard (RFC 2046, Chapter 5.2.2 - *Partial Subtype)* provides a mechanism for the splitting of large e-mails into multiple smaller parts (message/partial). This is done either by the sending software or by a mail server involved in the transmission process (Fig. 6 ① message/partial). The receiver must be capable of detecting the received split e-mail parts and assembling them into the original e-mail (Fig. 6 ②).

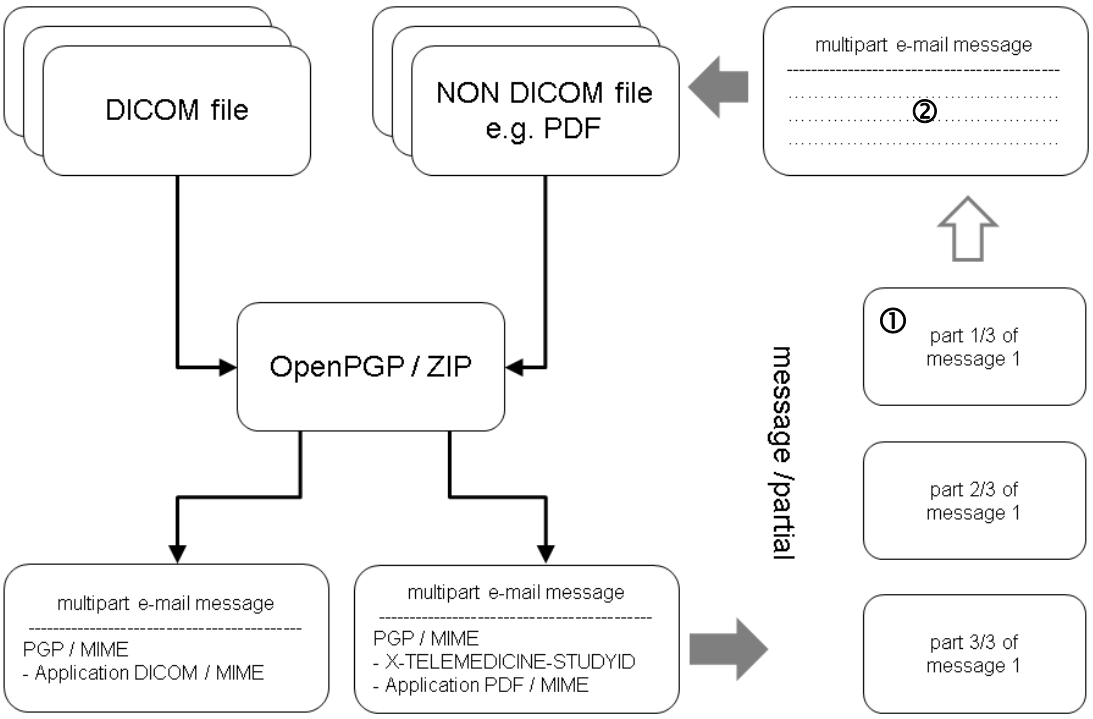

Fig. 6 – Data transfer overview

17.4 Mechanisms to verify the completeness of transmitted data

While sending data as DICOM E-MAIL, both sender and receiver must be able to verify the completeness of the transmitted data.

There are two principal ways to ensure this. The easiest would be to tell the receiver how many e-mails are expected and which part of the total transmission set was received. This can be achieved by adding a mail count to each transferred e-mail. (17.4.1 – Message Sets)

The second option is to return notify e-mails with information about the transferstatus to the sender (17.4.2 E-mail notification mechanisms).

17.4.1 Message sets (optional)

Used X-tags: *X-TELEMEDICINE-SETID*
X-TELEMEDICINE-SETPART
X-TELEMEDICINE-SETTOTAL

The sender adds additional X-tags to the e-mail MIME header, which can be interpreted by the receiver. This allows the receiver to verify whether associated e-mails have been received completely.

X-TELEMEDICINE-SETID = Unique ID to label a set of associated e-mails.

X-TELEMEDICINE-SETPART = The number of the current e-mail within the Message Set.

X-TELEMEDICINE-SETTOTAL = Total number of associated e-mails in a set.

These three X-tags group associated e-mails and allow the receiver to verify whether the complete set of e-mails has been received. The use of the X-TELEMEDICINE-SET is optional. In case the tag is used the following rules must apply:

1. The X-tags can be used in- and outside of the PGP/MIME containers (Fig. 7 – Labeling e-mails belonging together)
2. *X-TELEMEDICINE-SETID* and *X-TELEMEDICINE-SETPART* must be used in each e-mail.
3. *X-TELEMEDICINE-SETTOTAL* has to be present in the last e-mail of the set. It is optional for all other e-mails of the set.
4. X-tags should always be part of the encrypted e-mail body. In case they differ from unencrypted X-tags, a warning message should be issued.

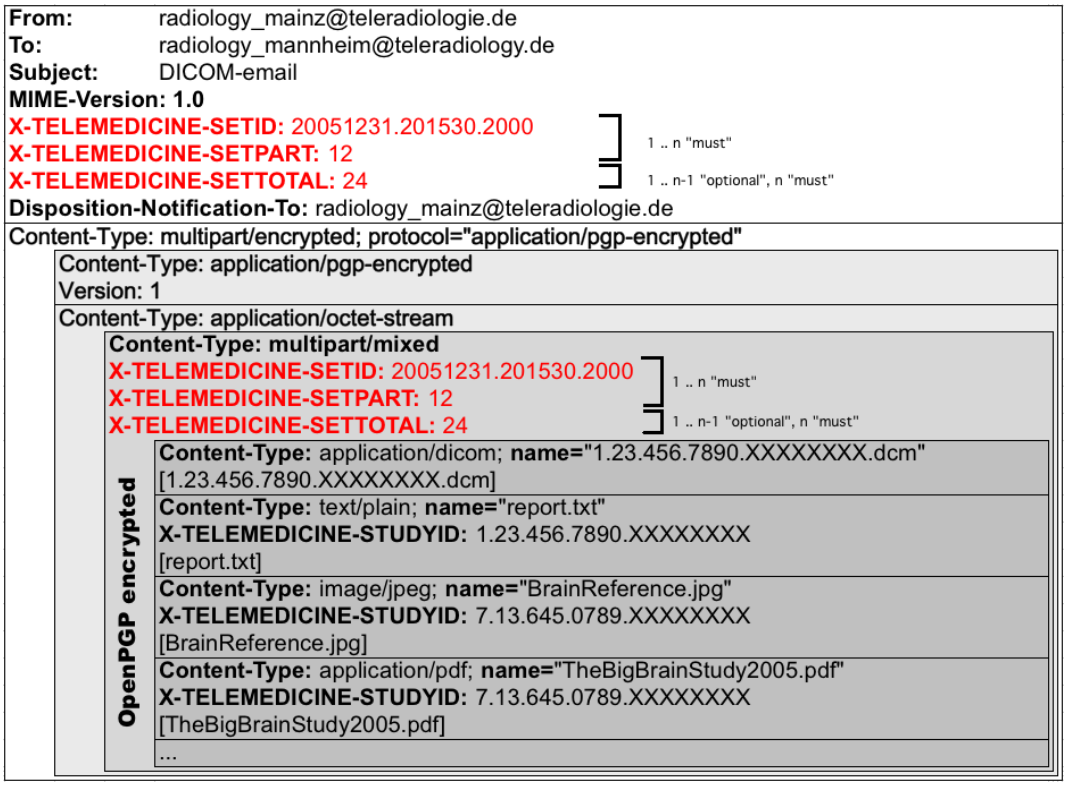

Fig. 7 – Labeling e-mails belonging together

The example in *Fig. 7 – Labeling e-mails belonging together* enables the receiver to evaluate the transmission status. This is only relevant for the receiver. It does not provide the sender with any new information. Upon receiving, the sender must be supplied with status information. This will be addressed in the following chapter 17.4.2 E-mail notification mechanisms.

17.4.2 E-mail notification mechanisms

On the sender side, the verification procedure for a complete data transmission consists of two interdependent steps (Step 1: Request for a receipt upon data reception, Step 2: Returning the requested receipt to the sender).

There are three different mechanisms for receiving notification e-mails. Each of these mechanisms returns a different kind of notification e-mail as shown in *Fig. 8 – E-mail notification mechanisms*.

1. Mechanism 1 (fall-back)
   The answer to a MIME conform request must always be a MIME conform e-mail notification as described in the RFC.
2. Mechanism 2 (obsolete)
   Answering a Tele-X-Standard conform request can either be done with a Tele-X-Standard conform e-mail notification or by using the fall-back of MIME conform e-mails notification.
3. Mechanism 3 (mandatory as of version 1.7)
   This mechanism is requested by adding X-tags to the MIME header. The notification will be sent as a Service Part e-mail of the type DISPOSITIONNOTIFICATION.

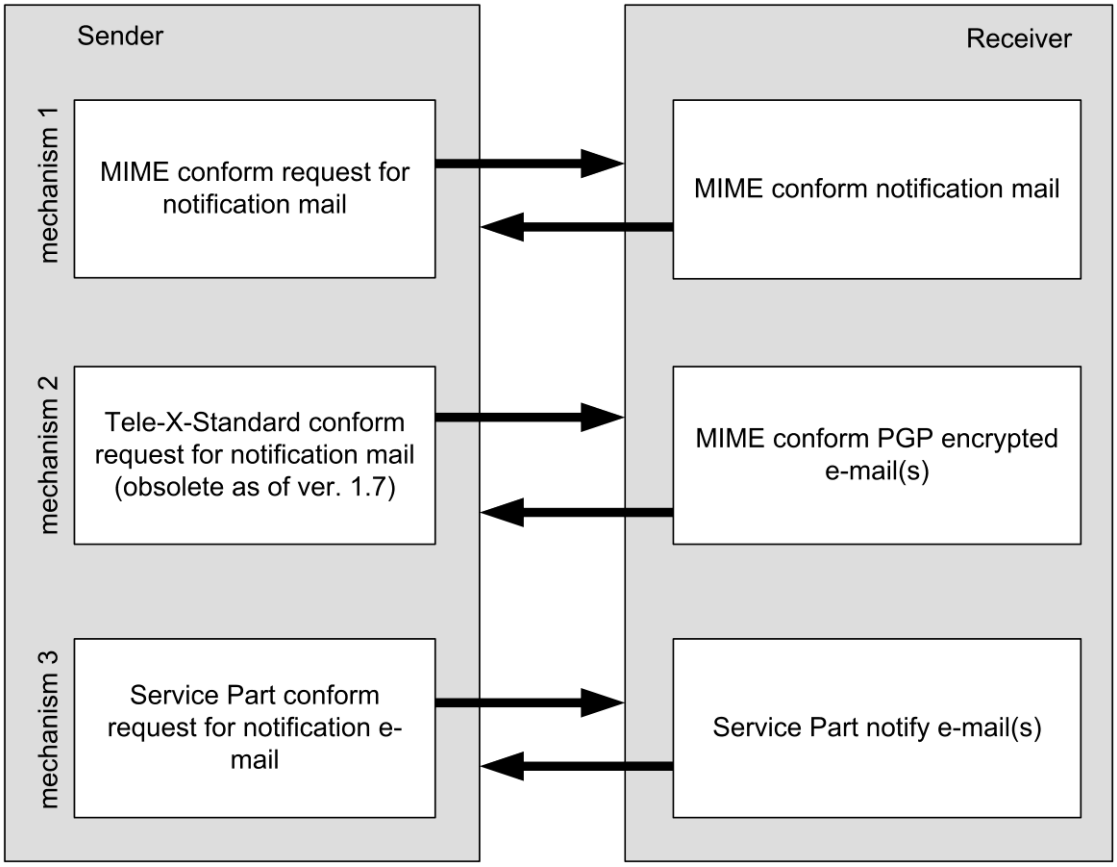

Fig. 8 – E-mail notification mechanisms

For any type of notification e-mail, the status upon reception is expressed by its status code (cf. chapter 24 Appendix Error Codes). In general, replying to notify request e-mails is optional. However, the implementation of the different mechanisms is obligatory as described in this Standard Recommendation.

As of version 1.7 of the Standard Recommendation, the mechanism for Tele-X-Standard conform notification e-mails (mechanism 2) is no longer supported. To ensure downward compatibility, the definition is included in this document.

17.4.2.1 E-mail notification mechanism 1 (mandatory)

Used MIME-Tags: *Message-ID*
Disposition-Notification-To

The e-mail notification mechanism 1 is the basic mechanism for requesting notification e-mails. It is also the fall-back to all other mechanisms described

below. Its support is mandatory. Returning the actual notification e-mails is optional.

Step 1: MIME conform request for notification e-mail

To verify the correct receipt of e-mails, the MIME-Standard provides the option to request notification e-mails (RFC 3798). The request (*Disposition-Notification-To: <notification address>*) is added to the e-mail MIME header. On receiver side, this triggers the mechanism to send a message disposition notification compliant with the RFC. The subsequent Fig. 9 shows a sample e-mail.

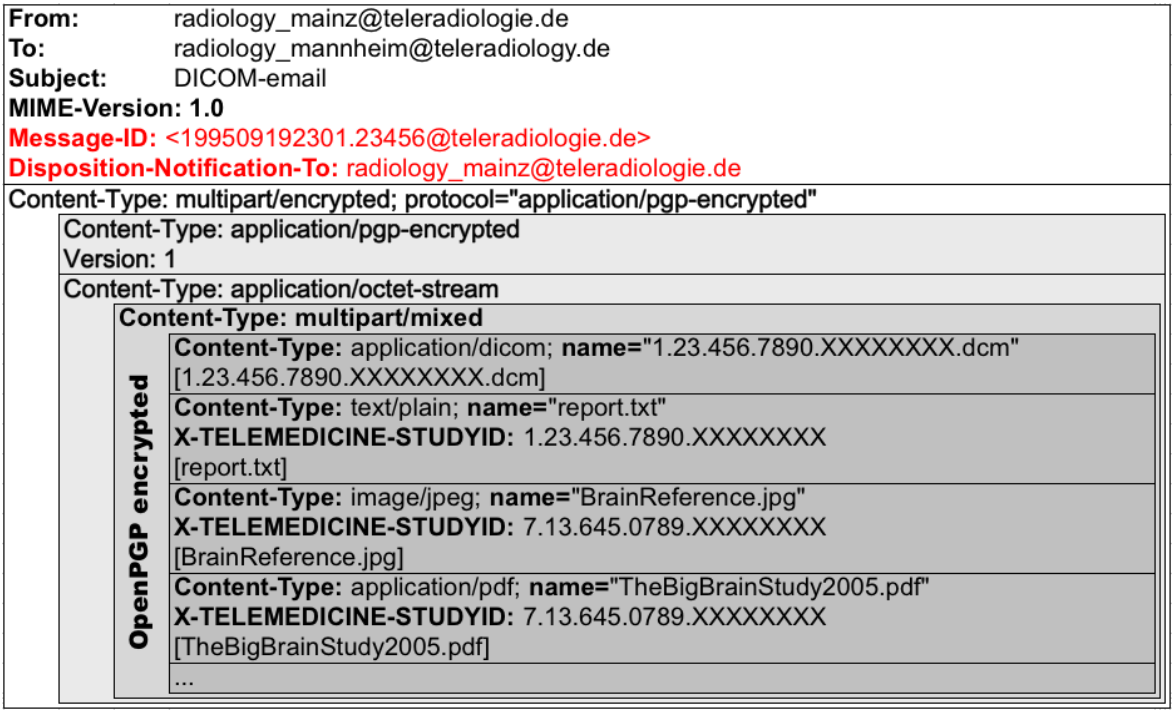

Fig. 9 – Request for confirmation of receipt compliant with RFC

As it is a basic mechanism, the request for notification is done for the entire DICOM E-MAIL message.

Multiple recipients for the resulting notification e-mail can be enumerated in a comma separated list as the value of the tag Disposition-Notification-To. One message per recipient only will be sent out as defined in the RFC 3798.

No special precautions for data protection purposes such as encryption or e-signature will be provided. To avoid this, mechanism 2 (obsolete) or mechanism 3 as described below may be used.

Step 2: MIME conform notification e-mail

The result of the request (Fig. 9 – Request for confirmation of receipt compliant with RFC) is a MIME conform reply e-mail as defined in RFC 3798. The e-mail is divided into a header and a „multipart/report“ body. The body consists of three parts:

- Part 1 – human readable (mandatory)
- Part 2 – machine readable (mandatory)
- Part 3 – reference to the original e-mail (optional)

The e-mail reply to a disposition notification request (*Disposition-Notification-To*) is always done by a “multipart/report” message.

Fig. 10 – Message disposition notification according to RFC 3798 shows a sample reply to the disposition notification request from Fig. 9. In this case, the return message indicates that the e-mail received previously contains a syntax error which has led to no relevant error in processing (Warning). The possible status messages are listed in chapter 17.4.2.1.1 Mechanism 1 – Disposition notification status messages.

```
Date: Mon, 1 Jan 2009 00:19:00 (EDT) -0400
From: radiology_mannheim@teleradiology.de
Message-Id: <2323423432019.12345@teleradiology.de>
Subject: Disposition notification
To: radiology_mainz@teleradiologie.de
MIME-Version: 1.0
Content-Type: multipart/report; report-type=disposition-notification;
boundary="RAA14128.773615765/teleradiology.de"

--RAA14128.773615765/teleradiology.de

 Optional text goes here

--RAA14128.773615765/teleradiology.de
content-type: message/disposition-notification

Reporting-UA: post.teleradiology.de; Mailprogram 1.1
Final-Recipient: rfc822; radiology_mainz@teleradiologie.de
Original-Message-ID: <199509192301.23456@teleradiologie.de>
Disposition: automatic-action/MDN-sent-automatically;displayed/warning
Warning: 1.2

--RAA14128.773615765/teleradiology.de--
```

Fig. 10 – Message disposition notification according to RFC 3798

17.4.2.1.1 Mechanism 1 – Disposition notification status messages

The MIME conform notification e-mail expects the status messages to be used as defined in RFC 3798. For this Standard Recommendation, only the two notification

types “Displayed” and “Deleted” are employed. Per definition, the notification types “No Errors” and “Warnings” are subsumed by the notification type “Displayed”. All processing errors are grouped in the status “Deleted”, including errors which either result in a retransfer of data or severe errors where data must not be re-sent.

The following combinations of status notifications and notification types are allowed:

1. no error ascertained
   disposition-field=Disposition:*automatic-action/MDN-sent-automatically*;**displayed**
   –[**no warning, error or failure field allowed**]

2. no error ascertained relevant for processing
   disposition-field=Disposition:*automatic-action/MDN-sent-automatically*;**displayed/warning**
   warning-field = "Warning" ":" *status –[**at least one warning field mandatory**]
   –[**no error or failure field allowed**]

3. Errors ascertained relevant for processing, resend e-mail
   disposition-field=Disposition:*automatic-action/MDN-sent-automatically*;**deleted/error**
   error-field = "Error" ":" *status –[**at least one error field mandatory**]
   warning-field = "Warning" ":" *status –[**warning field optional**]
   –[**no failure field allowed**]

4. Errors ascertained relevant for processing, do not resend e-mail
   disposition-field=Disposition:*automatic-action/MDN-sent-automatically*;**deleted**
   failure-field = "Failure" ":" *status –[**at least one failure field mandatory**]
   error-field = "Error" ":" *status –[**error field optional**]
   warning-field = "Warning" ":" *status –[**warning field optional**]

The wildcards (indicated as *status) contain the status code as defined in the Appendix (cf. Chapter 24 Appendix Error Codes).

Whether an error is marked as “Warning”,“Error” or “Failure” will be defined by the sender of the notification.

17.4.2.2 E-mail notification mechanism 2 (obsolete)

In order to overcome the weaknesses mentioned regarding mechanism 1 (17.4.2.1 E-mail notification mechanism 1 (mandatory)), mechanism 2 allows a separate request to confirm the receipt for each part of the encrypted PGP/MIME container. In addition, the notification e-mail can be returned as a PGP/MIME encrypted e-mail in case the recipient of the request e-mail for notification e-mails has been provided with a valid encryption key ID and the corresponding key is present on the receiver’s side.

Step 1: Tele-X-Standard conform request for notification e-mail

Used MIME-Tags: *Content-ID*
Disposition-Notification-To
Message-ID

Used X-tags: *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO*
X-TELEMEDICINE-DISPOSITION-NOTIFICATION-KEYID

This mechanism was valid for version 1.5 to 1.6.1. It results in multiple notification e-mails. Requesting mechanism 2 notify e-mails is initiated by using the MIME-Tag *DISPOSITION-NOTIFICATION-TO* in the e-mail MIME header and the following three tags in each part of the multipart/mixed section for which a notification mail is requested.

1. Content-ID
   MIME conform ID labeling e-mail parts with unique IDs. In contrast to the MIME standard, the Content-ID is mandatory for mechanism 2.
2. X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO:<e-mail address>
   The tag value contains the e-mail address to which a notification mail must be sent upon receipt of the specific part.
3. X-TELEMEDICINE-DISPOSITION-NOTIFICATION-KEYID:<GPG KeyID>
   The tag value contains the OpenPGP key that must be used to encrypt the notification mail for the specific part. The use of this tag is optional.

The notification addresses as well as the key IDs can be different for all parts. If these tags are used, it is mandatory to set a Content-ID conforming to RFC 2392. Due to the required global uniqueness of this ID, it is possible to map each e-mail that has been sent and any of its containing parts to a receipt. While the return e-mail address can result from either an Address- or KeyID-tag, the *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO* tag must be used if both are present.

The address set with *Disposition-Notification-To* does not have to match the one specified in *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO*. However, both must contain valid return e-mail addresses. If mechanism 2 is used, the return addresses from the encrypted part (*X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO*, *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-KEYID*) should primarily be used. The unencrypted address from *Disposition-Notification-To* should only be used as a fall-back.

In case mechanism 2 is used, it is mandatory to support mechanism 1 as well. This enables the reception of error messages even if the e-mail part could not be decrypted.

Fig. 11 – Content-related request for message disposition notification using the X-tags for key ID (long version) and return address

The example in Fig. 11 shows how to use the full extent of the *Tele-X-Standard conform request for notification e-mail* by defining the encryption key and the recipient of the notification e-mails for each part of the OpenPGP encrypted container.

Fig. 12 shows the use of *X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO* only.

Fig. 11 – Content-related request for message disposition notification using the X-tags for key ID (long version) and return address

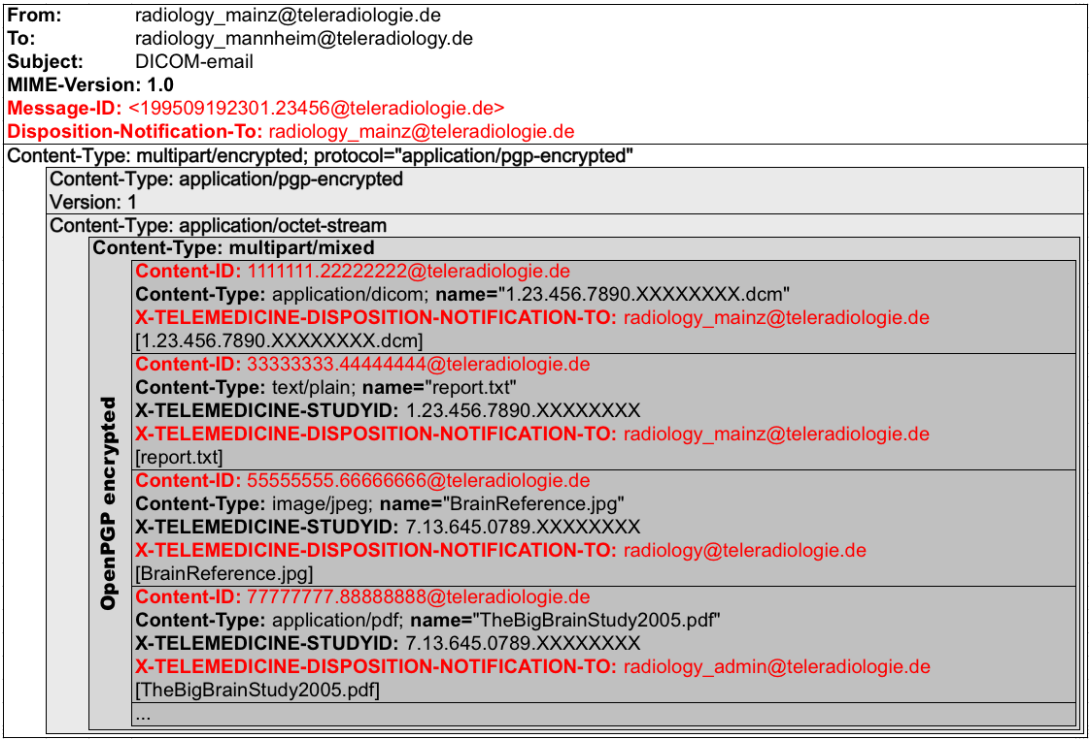

Fig. 12 – Content-related request for message disposition notification using the X-tag for return address only

Step 2*: Tele-X-Standard conform reply upon data reception*

Used X-tags: X-TELEMEDICINE-DISPOSITION-NOTIFICATION
X-TELEMEDICINE-ORIGINAL-CONTENT-ID

The reply to a message disposition notification request (Fig. 11) from the encrypted PGP/MIME container results in an encrypted confirmation e-mail.

The external e-mail container is built like a MIME conform encrypted e-mail. The encrypted part contains a MIME part with the content type „multipart/report“. The report type is “message/x-telemedicine-disposition-notification” which matches the content type of the report part.

This content type is defined analogously to „message/disposition-notification“ and contains the field X-TELEMEDICINE-ORIGINAL-CONTENT-ID.

Fig. 13 shows a reply example to the request from Fig. 11 – Content-related request for message disposition notification using the X-tags for key ID (long version) and return address.

```
From: radiology_mannheim@teleradiology.de
To: radiology_mainz@teleradiologie.de
Subject: Disposition notification
Date: Mon, 1 Jan 2009 15:15:35 +0100
MIME-Version: 1.0
Message-ID: <199509192301.23456@teleradiology.de >
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="--=_NextPart_000_0027_01BF27A0.9BE21980/teleradiology.de"

This is a multi-part message in MIME format.

----=_NextPart_000_0027_01BF27A0.9BE21980/teleradiology.de
Content-Type: application/pgp-encrypted

Version: 1

----=_NextPart_000_0027_01BF27A0.9BE21980/teleradiology.de
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: PGP 8.0.1*

%MIME-Version: 1.0
%Content-Type: multipart/report;
%              report-type=X-TELEMEDICINE-DISPOSITION-NOTIFICATION;
%boundary="RAA14128.773615765/teleradiology.de"
%
%--RAA14128.773615765/teleradiology.de
%
% Here is the spot for additional optional text%
%--RAA14128.773615765/teleradiology.de
%content-type: message/X-TELEMEDICINE-DISPOSITION-NOTIFICATION
%
%Reporting-UA: post.teleradiology.de; mail program 1.1
%Final-Recipient: rfc822; radiology_mainz@teleradiologie.de
%X-TELEMEDICINE-ORIGINAL-CONTENT-ID: 1111111.2222222@teleradiologie.de
%Disposition: automatic-action/MDN-sent-automatically;displayed/warning
%Warning: 1.2
%
```

```
%--RAA14128.773615765/teleradiology.de--

-----END PGP MESSAGE-----

----=_NextPart_000_0027_01BF27A0.9BE21980/teleradiology.de--
```

* Lines starting with % are PGP/GnuPG encrypted.

Fig. 13 – Confirmation e-mail as a reply to a request from the PGP/MIME Container

17.4.2.3 E-mail notification mechanism 3

Starting with version 1.7, this new mechanism for notification e-mails has been introduced as a replacement for the e-mail notification mechanism 2 (17.4.2.2). Just as the other two mechanisms 1 and 2, it works in two steps.

Step 1: Service Part conform request for notification e-mail

Used MIME Tags: Message-ID
Disposition-Notification-To
Content-ID

Used X-Tags: *X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-TO*
X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-KEYID

This mechanism is an advancement of the obsolete Tele-X-Standard conform request for a notification e-mail (17.4.2.2 E-mail notification mechanism 2 (obsolete)). As its predecessor, the service part conform request for a notification e-mail results in multiple notification e-mails.

For using this mechanism, the following three tags must be used in each of the parts of the multipart/mixed section in order to request a notification e-mail for each of the parts:

a. Content-ID
   MIME conform ID labeling of e-mail parts with unique IDs. In contrast to the MIME standard, the Content-ID is mandatory for mechanism 2.
b. X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-TO: <e-mail address>
   The tag value describes the receiver's e-mail address to which the notification mail must be sent upon receipt of the specific part.
c. X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-KEYID: <GPG KeyID>

The tag value describes the OpenPGP key that must be used to encrypt the notification mail for the specific part.

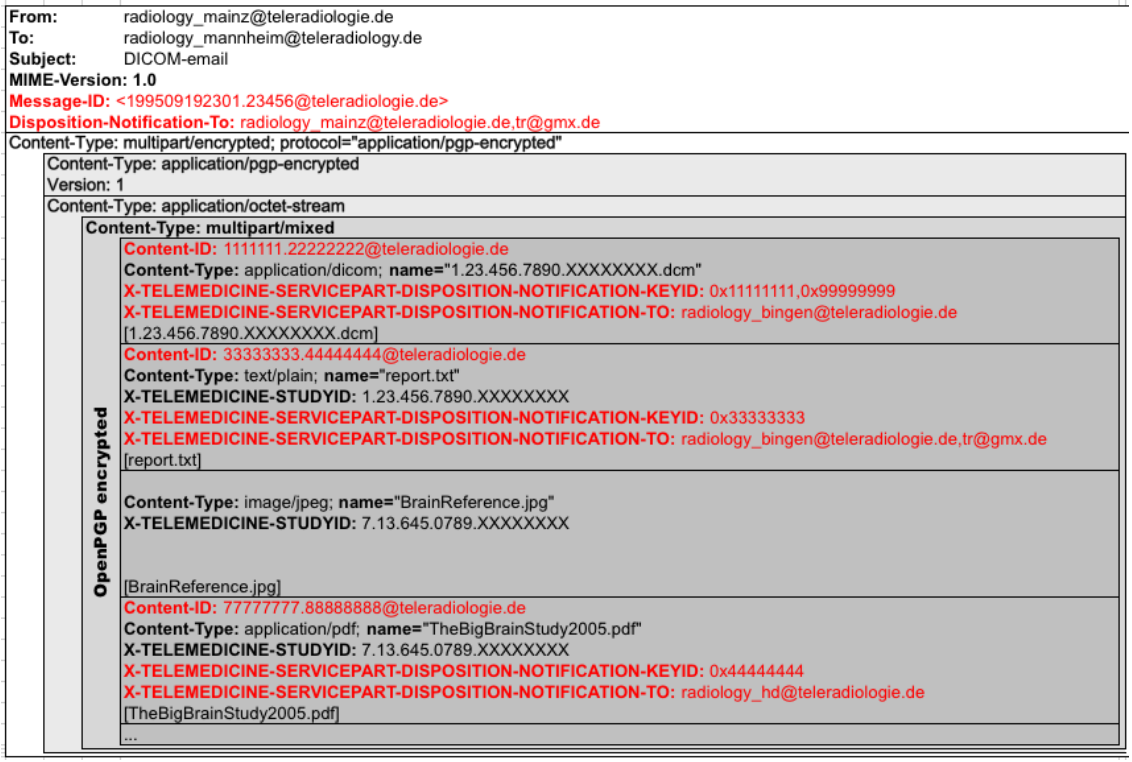

Fig. 14 – Example e-mail for a mechanism 3 notification request

Step 2: Service Part conform notification e-mail

Used MIME Tags: *Disposition-Notification-To*
Content-ID

Used X-Tags: *X-TELEMEDICINE-SERVICEPART:DISPOSITIONNOTIFICATION*
X-TELEMEDICINE-VERSION: x.y.z

The reply e-mail for mechanism 3 is a Service Part e-mail, which is defined in 18.3.5 Service Part – DISPOSITIONNOTIFICATION. Per recipient, only one notification e-mail is permitted. Therefore, if one e-mail contains multiple parts, the notifications of all parts must be collected per recipient and be returned within a single Service Part message. The notification should never contain responses to parts not requested by a specific recipient.

If multiple key IDs are defined, multi-encryption must be used for the notification. Therefore, if the same recipient address has been defined in more than one part, the notification e-mail must contain the corresponding XML structure for multiple parts as described in 18.3.5 Service Part – DISPOSITIONNOTIFICATION.

The corresponding disposition notification status messages are defined in chapter 18.3.5.1 Mechanism 3 – Disposition notification status messages.

The example Fig. 14 results in three Service Part Notification e-mails as displayed below in Fig. 15 - Fig. 17.

The e-mail in Fig. 15 must be multi-encrypted using the keys for number 0x11111111, 0x99999999 and 0x33333333. E-mail number two (Fig. 16) must be encrypted using key number 0x33333333. The last e-mail (Fig. 17) must be encrypted using key number 0x44444444.

```
From: radiology_mannheim@teleradiology.de
To: radiology_bingen@teleradiologie.de
Message-ID: <31175293.51336059080045@PC-123>
Subject: Mechanism type 3 notification e-mail
Date: Sun, 13 May 2013 12:39:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION
X-TELEMEDICINE-VERSION: 1.7.0

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40

%Content-Type: multipart/mixed;
%boundary="----=_Part_894967_718503476.1368448709295"
%
%------=_Part_894967_718503476.1368448709295
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="DISPOSITIONNOTIFICATION" timestamp="2013-05-13T12:38:29Z">
%    <MessageID>31175293.51336059080045@PC-123</MessageID>
%    <Notification>
%        <ContentID>1111111.22222222@teleradiologie.de</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%    <Notification>
%        <ContentID>33333333.44444444@teleradiologie.de</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%</ServicePart>
------=_Part_894967_718503476.1368448709295--
```

* Lines starting with % are PGP/GnuPG encrypted.

Fig. 15 – ServicePart Notification e-mail for the parts with ContentID 1111111.22222222@teleradiologie.de and 33333333.44444444@teleradiologie.de

```
From: radiology_mannheim@teleradiology.de
To: tr@gmx.de
Message-ID: <31175293.51336059080045@PC-123>
Subject: Mechanism type 3 notification e-mail
Date: Sun, 13 May 2013 12:39:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION
X-TELEMEDICINE-VERSION: 1.7.0

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40

%Content-Type: multipart/mixed;
%boundary="----=_Part_894967_718503476.1368448709295"
%
%------=_Part_894967_718503476.1368448709295
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="DISPOSITIONNOTIFICATION" timestamp="2013-05-13T12:38:29Z">
%    <MessageID>31175293.51336059080045@PC-123</MessageID>
%    <Notification>
%        <ContentID>33333333.44444444@teleradiologie.de</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%</ServicePart>
------=_Part_894967_718503476.1368448709295--
```

* Lines starting with % are PGP/GnuPG encrypted.

Fig. 16 – ServicePart Notification e-mail for the part with ContentID 33333333.44444444@teleradiologie.de

```
From: radiology_mannheim@teleradiology.de
To: radiology_hd@teleradiologie.de
Message-ID: <31175293.51336059080045@PC-123>
Subject: Mechanism type 3 notification e-mail
Date: Sun, 13 May 2013 12:39:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION
X-TELEMEDICINE-VERSION: 1.7.0

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40
```

```
%Content-Type: multipart/mixed;
%boundary="----=_Part_894967_718503476.1368448709295"
%
%------=_Part_894967_718503476.1368448709295
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="DISPOSITIONNOTIFICATION" timestamp="2013-05-13T12:38:29Z">
%    <MessageID>31175293.51336059080045@PC-123</MessageID>
%    <Notification>
%        <ContentID>77777777.88888888@teleradiologie.de</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%</ServicePart>
------=_Part_894967_718503476.1368448709295—
```

* Lines starting with % are PGP/GnuPG encrypted.

Fig. 17 – ServicePart Notification e-mail for the part with ContentID 77777777.88888888@teleradiologie.de

17.4.2.3.1 Conditions requiring the use of the fall-back mechanism

The fall-back mechanism (17.4.2.1 E-mail notification mechanism 1 (mandatory)) must be used under the following circumstances:

1. The encrypted container cannot be decrypted for any reason.
2. The client does not understand the X-tags used for mechanism 3.
3. The content or structure of the encrypted container does not match the description of mechanism 3 (17.4.2.3 E-mail notification mechanism 3).
4. One or more of the encryption keys is/are unknown or not available to the sender of the notification e-mails. In this case, the sender must return all notification e-mails as unencrypted notification mails to all recipients as defined in the MIME tag Disposition-Notification-To for mechanism 1 (please note that the recipients as defined by the X-tags of mechanism 3 must be ignored).

18 Service Part e-mails

The goal of Service Part e-mails is to extend the existing @GIT Teleradiology standard recommendation (versions 1.1 and 1.5) by adding the option to support workflows across vendors. The communication of Service Parts follows the rules of the previously used standards (cf. Chapter 21 – *Further applicable documents*).

The following scenarios are supported by Service Part e-mails:

1. Communication of data required for time constancy and function testing according to DIN 6868-159.
2. Exchange and management of PGP/GnuPG key data.
3. Exchange and management of address data.
4. Exchange and management of contact data.
5. Return of status messages.

18.1 Basic requirements for all Service Part e-mails

1. All Service Part e-mails must be encrypted and signed complying with to RFC (RFC 3156 and RFC 1847), in accordance with the mechanism defined by the @GIT Teleradiology Standard Recommendation (version 1.5+, cf. Chapter 16 Digital signature (PGP/MIME)).
   Recommendation: In order to avoid misuse by unauthorized e-mails, the receiver should maintain a white list (based on e.g. key IDs) before processing Service Part e-mails.
2. The content type of XML data must be text/xml.
3. The character set must be UTF-8 in the MIME header as well as in the XML structure.
4. Service Parts must be sent as multipart/mixed e-mails.
5. Multiple Service Parts can be sent in one combined Service Part e-mail. Such an e-mail will be marked by the X-tag X-TELEMEDICINE-SERVICEPART:MULTIPART (Version 1.7+).
6. The ServicePart element must contain the timestamp attribute in the format of YYYY-MM-DDThh:mm:ssZ (ISO 8601), (Version 1.7+).
7. All Service Part e-mails must indicate the supported @GIT Teleradiology Standard Recommendation version by using X-TELEMEDICINE-VERSION:x.y.z (3-digit).

8. Replies to Service Part e-mails must be sent as Service Part notification e-mails (17.4.2.3 E-mail notification mechanism 3) with the possibility of a fall-back to MIME conform notification e-mails (17.4.2.1 E-mail notification mechanism 1 (mandatory)). The circumstances of a permitted fall-back are described in chapter 17.4.2.3.1 Conditions requiring the use of the fall-back mechanism. In order to maintain backward compatibility, the existing error codes (cf. Chapter 24 Appendix Error Codes) have been extended accordingly (Version 1.7+).

The following table (Fig. 18 – XML value definitions for Service Parts) gives an overview of value definitions to be used in the XML structure of the Service Part e-mails.

Value	**Format**	**Version**
XML elements	Identical opening / closing tags in CamelCase with case insensitive interpretation	1.7+
XML attributes	Lower case with case insensitive interpretation	1.7+
timestamp	YYYY-MM-DDThh:mm:ssZ (ISO 8601)	1.7+
ContactID	UUID (rfc4122)	1.7+
ConnectionID	UUID (rfc4122)	1.7+
GPGKeyID	The last 16 digits of the fingerprint (long version)	1.7+
Character set for any value	UTF – 8	1.7+

Fig. 18 – XML value definitions for Service Parts

18.2 Structure of Service Part e-mails

All Service Part e-mails contain the X-tag *X-TELEMEDICINE-SERVICEPART* in the (unencrypted) e-mail MIME header which describes the requested action. This allows filtering of e-mails without processing the content of the XML document. (Example: *X-TELEMEDICINE-SERVICEPART:ADDRESSUPDATE*)

Service Part e-mails can contain either just a single Service Part (Fig. 19 – Service Part XML structure) (Version 1.6+) or multiple Service Parts (Version 1.7+) as shown in *22.1 Multipart Service Part Message using Notify Mechanism 3*. For downward compatibility, Single-Service Part e-mails need to be marked with the X-tag X-TELEMEDICINE-SERVICEPART:<Servicepart name>, Multi-Service Part e-mails will be marked with X-TELEMEDICINE-SERVICEPART:MULTIPART.

The following Fig. 19 – Service Part XML structure shows the basic layout of a Service Part XML structure. Its content varies with the specific Service Parts as described in chapter 18.3 Defined Service Parts.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="" action="" timestamp="YYYY-MM-DDThh:mm:ssZ>
...
</ServicePart>
```

Fig. 19 – Service Part XML structure

The *name* attribute defines the Service Part. The *action* attribute contains a detailed action (Example: name="addressupdate" action="remove"). The timestamp contains the exact time of the issuing of a Service Part. *Name* and *timestamp* are mandatory for all Service Parts (Version 1.7+). The occurance of *action* depends on the specific Service Part as defined by *name*.

18.3 Defined Service Parts

The following table (Fig. 20 – Overview of the defined Service Parts) lists all defined Service Parts. Additionally, the version of the DICOM E-MAIL Standard Recommendation is specified with their first appearance.

Service Part Name	Attribute	Type	Version
ADDRESSUPDATE	CLEAN	M	1.7+
	GET	M	1.7+
	REMOVE	M	1.6+
	SET	M	1.6+
CONNECTIONUPDATE	GET	O	1.7+
CONTACTUPDATE	CLEAN	M	1.7+
	GET	M	1.7+
	REMOVE	M	1.7+
	SET	M	1.7+
DISPOSITIONNOTIFICATION	-	M	1.7+
KEYUPDATE	CLEAN	M	1.7+
	GET	M	1.7+
	REMOVE	M	1.6+
	SET	M	1.6+
PROTOCOL		M	1.6+
QUERY	FIND	O	1.7+
	RESULT	O	1.7+
TESTTRANSFER		M	1.6+

Fig. 20 – Overview of the defined Service Parts

18.3.1 Legend of the Service Part structure tables

The structure of the Service Part XML content is printed like the tables in the DICOM standard.

The general structure is as follows:

<Element name>

- *<attribute>* (same level as the element above)

> *<attribute>* (plus one additional level for each preceding **>**)

The table header names used in this chapter represent the following:

Name = Element or attribute name; these names can be preceded by **-** or **>** depending on the correct level.

Type = (M)andatory, (C)onditional, (O)ptional

Multiplicity = Allowed number of repetition

Description = Description text

Version = Version of the DICOM E-MAIL standard recommendation

18.3.2 Service Part ADDRESSUPDATE

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: ADDRESSUPDATE*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

The Service Part ADDRESSUDATE enables the handling of data essential for a successful data exchange using DICOM E-MAIL. This Service Part provides the following actions:

Action	Chapter	Introduced Version	Last Change
CLEAN	18.3.2.1	1.7+	
GET	18.3.2.2	1.7+	
REMOVE	18.3.2.3	1.6+	1.7
SET	18.3.2.4	1.6+	1.7

Fig. 21 – ADDRESSUPDATE actions

Starting with version 1.7, the element ID became mandatory. Until version 1.6.1, the use of that element had been optional.

18.3.2.1 Action “CLEAN”

Version: 1.7+

The action “CLEAN” deletes all redundant address data which has not been defined by the element "KeepConnectionID"..

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="ADDRESSUPDATE" action="CLEAN" timestamp="YYYY-MM-DDThh:mm:ssZ">
<Connection>
 <KeepConnectionID />
</Connection>
</ServicePart>
```

Fig. 22 – XML example ADDRESSUPDATE CLEAN

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"ADDRESSUPDATE"
- action	M	1	“CLEAN”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> Connection	M	1	
>> KeepConnectionID	M	1-n	Defining the Connection IDs of the data sets to be retained while all other connections will be deleted.

Fig. 23 – XML structure ADDRESSUPDATE CLEAN

18.3.2.2 Action “GET”

Version: 1.7+

The action “GET” retrieves address data of a known communication path, identified by the element “ID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="ADDRESSUPDATE" action="GET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <Connection>
  <ID />
 </Connection>
</ServicePart>
```

Fig. 24 – XML example ADDRESSUPDATE GET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"ADDRESSUPDATE"
- action	M	1	“GET”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"

> Connection	M	1	
>> ID	M	1-n	Unique communication path ids for which an update has been requested

Fig. 25 – XML structure ADDRESSUPDATE GET

18.3.2.3 Action “REMOVE”

Version: 1.6+

The action “REMOVE” deletes address data of a specific communication path, identified by the element “ID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="ADDRESSUPDATE" action="REMOVE" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <Connection>
  <ID />
 </Connection>
</ServicePart>
```

Fig. 26 – XML example ADDRESSUPDATE REMOVE

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"ADDRESSUPDATE"
- action	M	1	“REMOVE”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> Connection	M	1	
>> ID	M	1	Unique id of the communication path to be removed

Fig. 27 – XML structure ADDRESSUPDATE REMOVE

18.3.2.4 Action “SET”

Version: 1.6+

The action “SET” adds or updates address data of a specific communication path, identified by the element “ID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="ADDRESSUPDATE" action="SET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <Connection>
  <ID />
  <DisplayConnectionName />
  <Mailserver />
  <Port />
  <EMailAddress />
  <GPGKeyID />
```

```
 </Connection>
</ServicePart>
```

Fig. 28 – XML example ADDRESSUPDATE SET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"ADDRESSUPDATE"
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> Connection	M	1	
>> ID	M	1	Unique communication path id
>> DisplayConnectionName	M		Connection name as seen by the user
>> Port	O	1	SMTP mail server port
>> EMailAddress	M	1	E-mail address of recipient
>> GPGKeyID	M	1	Long GPG key ID of the key to be used for this connection id

Fig. 29 – XML structure ADDRESSUPDATE SET

18.3.3 Service Part – CONNECTIONUPDATE

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: CONNECTIONUPDATE*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

This Service Part is a combination of ADDRESSUPDATE, KEYUPDATE and CONTACTUPDATE.

18.3.3.1 Action “GET”

Version: 1.7+

The action “GET” retrieves a complete communication path, identified by the element “ID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="CONNECTIONUPDATE" action="GET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <Connection>
  <ID />
 </Connection>
 <GPGKeyID />
 <ContactID />
</ServicePart>
```

Fig. 30 – XML example CONNECTIONUPDATE GET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"CONNECTIONUPDATE"

- action	M	1	“GET”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> Connection	M	1	
>> ID	M	0-n	Unique IDs of the communication paths for which a data update is requested; in case of no ID the recipient returns all available connection data.
> GPGKeyID	O	1-n	OpenPG key IDs (long version) for which a data update is requested
> ContactID	O	1-n	Unique Contact IDs for which a data update is requested

Fig. 31 – XML structure CONNECTIONUPDATE GET

18.3.4 Service Part – CONTACTUPDATE

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: CONTACTUPDATE*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

Version 1.7+

Starting with version 1.7, the standard recommendation defines the values of a contact (Fig. 40 – XML structure CONTACTUPDATE SET). The Service Part CONTACTUPDATE offers the option to exchange and update these contacts.

Action	Chapter	Introduced Version	Last Change
CLEAN	18.3.4.1	1.7+	
GET	18.3.4.2	1.7+	
REMOVE	18.3.4.3	1.6+	1.7
SET	18.3.4.4	1.6+	1.7

Fig. 32 – CONTACTUPDATE actions

18.3.4.1 Action “CLEAN”

Version 1.7+

The action “CLEAN” deletes all redundant contact data which has not been defined by the element "KeepConnectionID".

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="CONTACTUPDATE" action="CLEAN" timestamp="YYYY-MM-DDThh:mm:ssZ">
  <KeepContactID />
</ServicePart>
```

Fig. 33 – XML example CONTACTUPDATE CLEAN

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"CONTACTUPDATE"
- action	M	1	“CLEAN” – defined action
- timestamp	M	1	ISO 8601 timestamp “YYYY-MM-DDThh:mm:ssZ”
> KeepContactID	O	0-n	Contact ID(s) defining the data set(s) that are excluded from deletion. All other data sets should be deleted. If no KeepContactID is provided, all contact data sets will be deleted.

Fig. 34 – XML structure CONTACTUPDATE CLEAN

18.3.4.2 Action “GET”

Version 1.7+

The action “GET” retrieves contact data of a known communication path, identified by the element “ContactID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="CONTACTUPDATE" action="GET" timestamp="YYYY-MM-DDThh:mm:ssZ">
  <ContactID />
</ServicePart>
```

Fig. 35 – XML example CONTACTUPDATE GET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"CONTACTUPDATE"
- action	M	1	“GET” – defined action
- timestamp	M	1	ISO 8601 timestamp “YYYY-MM-DDThh:mm:ssZ”
> ContactID	M	1-n	Contact ID(s) defining the data set(s) that are requested by the sender.

Fig. 36 – XML structure CONTACTUPDATE GET

18.3.4.3 Action „REMOVE“

Version 1.7+

The action “REMOVE” deletes address data of a specific communication path, identified by the element “ContactID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="CONTACTUPDATE" action="REMOVE" timestamp="YYYY-MM-DDThh:mm:ssZ">
  <ContactID />
</ServicePart>
```

Fig. 37 – XML example CONTACTUPDATE REMOVE

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"CONTACTUPDATE"
- action	M	1	“REMOVE” – defined action
- timestamp	M	1	ISO 8601 timestamp “YYYY-MM-DDThh:mm:ssZ”
> ContactID	M	1-n	Contact ID(s) defining the data set(s) that should be deleted.

Fig. 38 – XML structure CONTACTUPDATE REMOVE

18.3.4.4 Action “SET”

Version 1.7+

The action “SET” adds or updates contact data of a specific communication path, identified by the element “ContactID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="CONTACTUPDATE" action="SET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <ContactID />
 <ConnectionID />
 <Comment />
 <Role />
 <Unit />
 <Specialty />
 <Title />
 <Sex />
 <GivenName />
 <Name />
 <ContactInformation order="integer value">
  <RealLifeEMailAddress />
   <Fax />
   <Mobile />
   <Phone />
   <Pager />
   <Website />
   <Comment />
 </ContactInformation>
 <Address order="integer value">
  <Comment />
  <Street_1 />
  <Street_2 />
  <City />
  <ZipCode />
  <State />
  <Country />
 </Address>
 <RelatedID />
 <ContainerID />
 <Custom identifier="text" version="text">
  <!-- room for private elements //-->
 </Custom>
</ServicePart>
```

Fig. 39 – XML example CONTACTUPDATE SET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"CONTACTUPDATE"

- action	M	1	“SET” – defined action
- timestamp	M	1	ISO 8601 timestamp “YYYY-MM-DDThh:mm:ssZ”
> ContactID	M	1	Contact ID defining the data set to which the action should be applied
> ConnectionID	O	1-n	The original connection ID
> Comment	O	1	Field for additional description
> Role	O	1-n	Role defines the occupation. Standardized values are MEDICAL (e.g. nurses, technicians etc.) TECHNICAL (e.g. IT, Biomedical etc.) ADMINISTRATION (e.g. ward secretary, administration) DOCTOR (M.D. only)
> Unit	M	1	Unit defines the place of the attached data in the organisation hierarchy. Possible values are: PERSON, DEPARTMENT, HOSPITAL, ORGANIZATION
> Speciality	O	1-n	Additional text field for providing more detailed information about the level of expertise (e.g. orthopaedic implants; spine surgery)
> Title	O	1	Academic title(s) (e.g. Prof. Dr. Dr., MD, RCFR)
> Sex	M	1	[m (male) / f (female) / o (other)]
> Name	M	1	Name according to the defined Unit (s.o.) (e.g. Family name, department name)
> ContactInformation	O	1-n	Tag containing all contact information not related to a postal address.
- order	O	1	Non unique integer value setting the order of multiple ContactInformation tags.
>> RealLifeEMailAddress	O	1	E-mail address which is not used for DICOM E-MAIL
>> Fax	O	1	Fax number (+[CountryCode]-[area code]-[subscribers number]-[extension])
>> Mobile	O	1	Mobile number (+[CountryCode]-[area code]-[subscribers number])
>> Phone	O	1	Telephone number (+[CountryCode]-[area code]-[subscribers number])
>> Pager	O	1	[pager number] or +[CountryCode]-[area code]-[subscribers number]-[extension]
>> Website	O	1	[Protocol]://url (e.g. https://www.uni-mainz.de)
>> Comment	O	1	Text field for additional information
> Address	O	1-n	Tag containing all postal address related information
- order	O	1	Non unique integer value setting the order of multiple Address tags.
>> Comment	O	1	Text field for additional information
>> Street_1	O	1	Postal address field #1
>> Street_2	O	1	Postal address field #2
>> City	O	1	Postal address field containing the city name
>> ZipCode	O	1	Postal address field containing the zip code
>> State	O	1	Postal address field containing the state’s name

>> Country	O	1	Postal address field containing the country name
> RelatedID	O	1-n	Contact ID of a related data set
> ContainerID	O	1	ID of the organizational unit from the level above
> Custom			Tag containing all added custom tags
- identifier	O	1	Text
- version	O	1	Text
>> Private Tags	O	1-n	any tag

Fig. 40 – XML structure CONTACTUPDATE SET

18.3.5 Service Part – DISPOSITIONNOTIFICATION

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

Version 1.7+

This Service Part is the equivalent to the MIME conform notification e-mail and can be addressed as Service Part notification e-mail. It is used for e-mail notification mechanism 3 (17.4.2.3). The Service Part DISPOSITIONNOTIFICATION is the replacement for the notification e-mail used in e-mail notification mechanism 2 (17.4.2.2). This Service Part is the standard notification e-mail type as of version 1.7. It can only be replaced by a MIME conform notification e-mail under the circumstances described in 17.4.2.3.1 Conditions requiring the use of the fall-back mechanism.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="DISPOSITIONNOTIFICATION" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <MessageID />
 <Notification>
  <ContentID />)
  <DispositionField />
  <Response>
   <ErrorCode />
   <Comment />
  </Response>
 </Notification>
</ServicePart>
```

Fig. 41 – XML example DISPOSITIONNOTIFICATION

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"DISPOSITIONNOTIFICATION"
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> MessageID	M	1	The original message ID.
> Notification	M	1-n	
>> ContentID	C	1	The original content ID. Must be present if available.

			May be missing in case of encryption or if signature errors occur.
>> DispositionField	M	1	Defined values are: "displayed", "displayed/warning", "deleted/error" or "deleted", cf. 18.3.5.1 Mechanism 3 – Disposition notification status messages
>> Response	C	0-n	May not be present if the "DispositionField" value is "displayed".
>>> ErrorCode	M	1	The returned error code, cf. 18.3.5.1 Mechanism 3 – Disposition notification status messages
>>> Comment	O	1	Descriptive text corresponding to the ErrorCode e.g. by using the description from 24 Appendix Error Codes

Fig. 42 – XML structure DISPOSITIONNOTIFICATION

18.3.5.1 Mechanism 3 – Disposition notification status messages

In compliance with the status messages defined in RFC 3798 and as listed in chapter 17.4.2.1.1 Mechanism 1 – Disposition notification status messages, the Service Part DISPOSTIONNOTIFICATION uses the following status values (Fig. 43 – RFC 3798 conform status messages for Service Part DISPOSITIONNOTIFICATION):

Element	**Value**	**Meaning**
DispositionField	displayed	success
	displayed/warning	Success but minor problems occurred, which did not affect the successful data delivery
	deleted/error	Error occurred – data could not be delivered
	deleted	Error occurred – data could not be delivered
Error	Error code	Contains the corresponding error code as defined in chapter 24 Appendix Error Codes
Comment	Error description	Descriptive test corresponding to the error e.g. by using the description as defined in chapter 24 Appendix Error Codes

Fig. 43 – RFC 3798 conform status messages for Service Part DISPOSITIONNOTIFICATION

18.3.6 Service Part – KEYUPDATE

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: KEYUPDATE*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

Version 1.6+

The Service Part KEYUPDATE has been among the first Service Parts to be defined by the Recommendation for a Standardized Teleradiology Transmission

Format with version 1.6. It enables the exchange and update of public encryption certificates, essential for communication using DICOM E-MAIL. The following actions (Fig. 44 – KEYUPDATE actions) have been defined for this Service Part:

Action	Chapter	Introduced Version	Last Change
CLEAN	18.3.6.1	1.7+	
GET	18.3.6.2	1.7+	
REMOVE	18.3.6.3	1.6+	1.7
SET	18.3.6.4	1.6+	1.7

Fig. 44 – KEYUPDATE actions

18.3.6.1 Action “CLEAN”

Version 1.7+

The action “CLEAN” deletes all redundant public encryption certificates which have not been defined by the element "KeepGPGKeyID".

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="KEYUPDATE" action="CLEAN" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <KeepGPGKeyID />
</ServicePart>
```

Fig. 45 – XML example KEYUPDATE CLEAN

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"KEYUPDATE"
- action	M	1	“CLEAN”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> KeepGPGKeyID	M	0 - n	Defining the OpenPGP key IDs (long version) of the Public Keys to be retained while all other keys will be deleted.

Fig. 46 – XML structure KEYUPDATE CLEAN

18.3.6.2 Action “GET”

Version 1.7+

The action “GET” retrieves the public encryption certificate of a known GPG key, identified by the element “GPGKeyID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="KEYUPDATE" action="GET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <GPGKeyID />
</ServicePart>
```

Fig. 47 – XML example KEYUPDATE GET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"KEYUPDATE"
- action	M	1	“GET”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> GPGKeyID	M	1	Defining the OpenPG key IDs (long version) of the Public Keys that should be returned

Fig. 48 – XML structure KEYUPDATE GET

18.3.6.3 Action “REMOVE”

Version 1.6+

The action “REMOVE” triggers the deletion of the public encryption certificate of a specific GPG key, identified by the element “GpgKeyID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="KEYUPDATE" action="REMOVE" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <GPGKeyID />
</ServicePart>
```

Fig. 49 – XML example KEYUPDATE REMOVE

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"KEYUPDATE"
- action	M	1	“REMOVE”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> GPGKeyID	M	1	Long OpenPGP compatible Public key ID (long version) of the key that should be removed

Fig. 50 – XML structure KEYUPDATE REMOVE

18.3.6.4 Action “SET”

Version 1.6+

The action “SET” adds or updates the public encryption certificate for a specific gpg key, identified by the element “GPGKeyID”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="KEYUPDATE" action="SET" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <PublicKeyASCIIData />
</ServicePart>
```

Fig. 51 – XML example KEYUPDATE SET

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"KEYUPDATE"
- action	M	1	“SET”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> PublicKeyASCIIData	M	1	ASCII armored OpenPG compatible Public key

Fig. 52 – XML structure KEYUPDATE SET

18.3.7 Service Part – PROTOCOL (mandatory)

Used X-Tags: *X-TELEMEDICINE-SERVICEPART:PROTOCOL*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

Version 1.6+

The Service Part PROTOCOL is the reply to a request for a test transfer initiated by the Service Part “TESTTRANSFER” (18.3.9.1). The PROTOCOL Service Part contains information about the transmission of the requested test data set. As the PROTOCOL Service Part is issued by one client only, inconsistent time settings among the participants of the test transfer is irrelevant.

Each Service Part “PROTOCOL” must only contain one protocol XML structure. The Service Part “PROTOCOL” e-mail is sent automatically upon transmission of the test data. The XML structure of this Service Part is shown in Fig. 54 – XML structure Service Part PROTOCOL. The following Fig. 53 – XML example Service Part PROTOCOL displays an example of the structure defined in Fig. 53. The use of the Service Part “PROTOCOL” together with the Service Part “TESTTRANSFER” is demonstrated in chapter 19.1 Constancy tests conforming to DIN 6868-159.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="PROTOCOL" timestamp="YYYY-MM-DDThh:mm:ssZ">

  <TransmissionStatus />
  <TestDataSetID />

  <ObjectsSent>
    <Count />
  </ObjectsSent>

  <ObjectsReceivedConfirmed>
   <Count />
   <Time />
   <MailSize />
   <ObjectSize />
  </ObjectsReceivedConfirmed>

  <DataSender>
    <EMailAddress />
  </DataSender>

  <DataRecipient>
    <EMailAddress />
  </DataRecipient>

  <ProtocolRecipient>
   <EMailAddress />
  </ProtocolRecipient>

  <ErrorTimeOut />

  <DatagramMail EMailMessageID="">
   <ErrorID />
   <EMailContentID />
   <StartDateTime />
   <NotifyDateTime />
   <MailSize />
   <ObjectSize />
  </DatagramMail>

</ServicePart>
```

Fig. 53 – XML example Service Part PROTOCOL

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"PROTOCOL"
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
>TransmissionStatus	M	1	Possible states are COMPLETED, ABORTED
>TestDataSetID	M	1	ID of the transferred test-data set
ObjectsSent	M	1	
>>Count	M	1	Count of the transferred objects
>ObjectsReceivedConfirmed	M	1	Summary of all DatagramMail nodes
>>Count	M	1	Count of transferred and verified objects

>>Time	M	1	Transmission period (beginning with the transfer of the data package and ending with the last verification e-mail), specified in seconds
>>MailSize	M	1	Combined size of all verified e-mails , specified in bytes
>>ObjectSize	M	1	Combined size of all verified objects , specified in bytes
>DataSender	M	1	
>>EMailAddress	M	1	e-mail address of the test-data set sender
>DataRecipient	M	1	
>>EMailAddress	M	1	e-mail address of the test-data set receiver
>ProtocolRecipient	M	1	
>>EMailAddress	M	1	e-mail address of the protocol receiver
>ErrorTimeOut	M	1	Period, specified in seconds, after which receiving disturbances are regarded as errors.
>DatagramMail EMailMessageID=""	M	1-n	This block can be repeatedarbitrarily
>>ErrorID	C	1	In case of an error: The tag contains the error code according to the error code list. The element is optional in the absence of an error.
>>EMailContentID	O	1	ContentID (Multipart Mails)
>>StartDateTime	M	1	Data transmission start date and time (format yyyymmddhhmmss)
>>NotifyDateTime	M	1	Notification e-mail receiving date and time (format yyyymmddhhmmss)
>>MailSize	M	1	Total size of all transmitted e-mails, , specified in bytes
>>ObjectSize	M	1	Total size of all transmitted objects, , specified in bytes

Fig. 54 – XML structure Service Part PROTOCOL

18.3.8 Service Part – QUERY (optional)

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: QUERY*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

The Service Part “QUERY” allows to query a communication data repository, such as a directory service. The Service Part provides two actions (Fig. 55 – QUERY actions), one is used to query (18.3.8.1 Action “FIND”) a repository, and the other enables the repository to provide the initial sender with the results (18.3.8.1 Action “FIND”).

Action	**Chapter**	**Introduced Version**	**Last Change**
FIND	18.3.8.1	1.7+	
RESULT	18.3.8.2	1.7+	

Fig. 55 – QUERY actions

18.3.8.1 Action “FIND”

Version 1.7+

The action “FIND” initiates a query on the recipient’s side . The query is defined by data tags and the expected value. The XML structure allows logical operations, such as AND, OR, NOT. As in DICOM C-FIND, the query also defines the return values (element ResultSet). The following two figures (Fig. 56 – XML example QUERY, Fig. 57 – XML structure QUERY) show the XML structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="QUERY" action="FIND" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <Search maxCount=0 - n >
   <Filter>
    <DataTag modifier="NOT">[Value]</DataTag>
   </Filter>
 </Search>
 <ResultSet filtered="true/false">
  <ContactID />
  <ConnectionID expand="true/false" />
   <Address>
   <AddressDataTag 1 />
    <AddressDataTag 2/>
    ....
   </Address>
  </ResultSet>
</ServicePart>
```

Fig. 56 – XML example QUERY FIND

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"QUERY"
- action	M	1	“FIND” – defined action
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> Search	M	1	This tag defines the search filter part of that Service Part
- maxCount	O	1	Defines the max. count of reply data sets. 0 = unlimited
> >Filter	M	1-n	The “Filter” tag defines the values which must be found in the replied data sets. All “Filter” tags are combined with a logical OR.
>>>DataTag	O	1-n	The DataTag and its value define the search filter parts. All DataTags are combined through a logical AND.
- modifier	O	1	“NOT” – the modifier applies a logical NOT to the value of that specific DataTag.
> ResultSet	O	1	This tag defines which data fields are to be included in the replied data sets. In absence of this element, all possible fields will be returned.
- filtered	C	1	This modifier determines that (filtered=true) only data should be returned which immediately can be used for establishing a communication link through the mail server in use. “filtered=false” would return any data matching the filter criteria.

>> ContactID	M	1	ContactID must be returned
>> ConnectionID	M	1	ConnectionID must be returned
- expand	O	1	“true” – defines that any data available for a given ConnectionID should be returned, including the ASCII armored PGP keys.
>> Address	O	1	Providing an Address tag without any additional AddressDataTags will result in the return of all available address data fields.
>>>AdressDataTag 1 .. n	O	1	These tags define which data fields should be returned

Fig. 57 – XML structure QUERY FIND

18.3.8.2 Action “RESULT”

Version 1.7+

The action “RESULT“ transmits the response of a Service Part QUERY FIND action (18.3.8.1 Action “FIND”).

```
<?xml version="1.0" encoding="UTF-8"?>
 <ServicePart name="QUERY" action="RESULT" timestamp="YYYY-MM-DDThh:mm:ssZ">
  <ResultSet>
   <Result>
    <DataTag 1 />
    <DataTag 2 />
    ...
   </Result>
 </ResultSet>
</ServicePart>
```

Fig. 58 – XML example QUERY RESULT

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	" QUERY"
- action	M	1	“RESULT” – defined action
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"
> ResultSet	M	1	The “ResultSet” tag combines the results of multiple queries.
> >Result	O	1-n	The “Result” tag contains the results of one query.
>>>DataTag 1 .. n	O	1-n	The DataTag and its value define the search filter parts. All DataTags are combined through a logical AND.

Fig. 59 – XML structure QUERY RESULT

18.3.9 Service Part – TESTTRANSFER

Used X-Tags: *X-TELEMEDICINE-SERVICEPART:TESTTRANSFER*
X-TELEMEDICINE-SERVICEPART-VERSION: x.y.z

Used MIME-Tags: *Disposition-Notification-To*

The Service Part “TESTTRANSFER” has been defined to run a performance test between two communication partners, resulting in a predefined protocol (Fig. 54 – XML structure Service Part PROTOCOL) that must be returned as a Service Part e-mail (18.3.7 Service Part – PROTOCOL (mandatory)) to an arbitrary receiver. The Service Part defines the following actions (Fig. 60 – TESTTRANSFER actions). A practical use of the Service Parts TESTTRANSFER and PROTOCOL is described in chapter 19.1 Constancy tests conforming to DIN 6868-159.

Action	Chapter	Version	Last Change
QOSCHECK	18.3.9.1	1.6+	1.7

Fig. 60 – TESTTRANSFER actions

18.3.9.1 Action “QOSCHECK”

Version 1.6+

The action “QOSCHECK” initiates the transfer of a specific test data set as a constancy test according to DIN 6868-159. The test data set is defined by the element TestDataSetID. The XML structure also defines the receiver of the test data set (TestDataReceiver) and the recipient of the resulting protocol (ProtocolReceiver) as defined in Fig. 62 – XML structure TESTTRANSFER QOSCHECK.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServicePart name="TESTTRANSFER" action="QOSCHECK" timestamp="YYYY-MM-DDThh:mm:ssZ">
 <TestDataReceiver>
  <EMailAddress />
  <GPGKeyID />
 </TestDataReceiver>
 <ProtocolReceiver>
  <EMailAddress />
  <GPGKeyID />
 </ProtocolReceiver>
 <TestDataSetID />
 <ErrorTimeOut />
</ServicePart>
```

Fig. 61 – XML example TESTTRANSFER QOSCHECK

Name	Type	Multiplicity	Description
ServicePart	M	1	
- name	M	1	"TESTTRANSFER"
- action	M	1	“QOSCHECK”
- timestamp	M	1	ISO 8601 timestamp "YYYY-MM-DDThh:mm:ssZ"

> TestDataReceiver	M	1	
> >EMailAddress	M	1	E-mail address of the data receiver
>>GPGKeyID	M	1	The ID of the encryption certificate (long version) which should be used to encrypt the test data set.
> ProtocolReceiver	O	1	
> >EMailAddress	C	1	Protocol e-mail receiver address
>>GPGKeyID	M	1	The ID of the encryption certificate (long version) which should be used to encrypt the resulting Service Part PROTOCOL
>TestDataSetID	M	1	e.g. TESTDATASET_1 (cf. Appendix Test Data Set IDs)
>ErrorTimeOut	O	1	This element defines the time in [s] after which a test protocol will be issued and transmitted, regardless of the test transfer's status.

Fig. 62 – XML structure TESTTRANSFER QOSCHECK

19 Application Scenarios for Service Part e-mails

The scenarios in this chapter describe the use of different Service Parts. The following table (Fig. 63 – Service Part Scenarios) gives an overview of the scenarios described below and the type (M=Mandatory, O=Optional) if they must or can be supported.

Scenario	**Type**	**Version**
Constancy tests (DIN 6868-159)	M	1.6+
Exchange of address data (set)	M	1.6+
Exchange of address data (get)	M	1.7+
Exchange of address data (remove)	M	1.7+
Exchange of address data (clean)	M	1.7+
Exchange of key data (set)	M	1.6+
Exchange of key data (get)	M	1.7+
Exchange of key data (remove)	M	1.6+
Exchange of key data (clean)	M	1.7+
Exchange of connection data (get)	O	1.7+
Exchange of contact data (clean)	M	1.7+

Exchange of contact data (get)	M	1.7+
Exchange of contact data (remove)	M	1.7+
Exchange of contact data (set)	M	1.7+
Query and retrieve connection data	O	1.7+

Fig. 63 – Service Part Scenarios

19.1 Constancy tests conforming to DIN 6868-159

Service Parts: *TESTTRANSFER QOSCHECK*
DISPOSITIONNOTIFICATION
PROTOCOL

For most Telemedicine scenarios, it is vital that connections are reliable and constantly available. This is especially important in emergency scenarios. Therefore in Germany, the standard DIN 6868-159 has been established in Teleradiology. It describes the intervals of constancy tests for the availability and the bandwidth stability including the parameters that should be tested.

In principle, all tests expect a typical sender-receiver communication scenario. The sender transmits a test data set to the receiver. The data set receiver returns a notification of the transmission including the status back to the sender. For a constancy test, the sender of the test data set knows all parts (status, amount of data, gross data transmission time) necessary to issue a report for the transmission which can be distributed to the corresponding communication partner. In many real life scenarios either the receiver of the test data sets or a third party is the maintainer of the Teleradiology connection and responsible for its maintenance. Therefore, it is necessary that constancy tests can also be triggered either by the receiver of the data sets or by a third party.

Additionally the following requirements must be met:

1. The test data required for constancy tests must be taken from a data pool which is defined by communication partners.
2. All types of data can be used.
3. The tested communication is defined between two DICOM E-MAIL nodes.

A typical process is illustrated in the following flowchart (Fig. 64 – Sample process: constancy test by Service Part trigger e-mail). It shows a potential process for testing the transfer speed between the communication partners B and C. Administrator A initiates the transfer of test data between B and C (Fig. 65 –

Example Service Part trigger e-mail). The communication protocol is transmitted by the communication partner B to the administrator A.

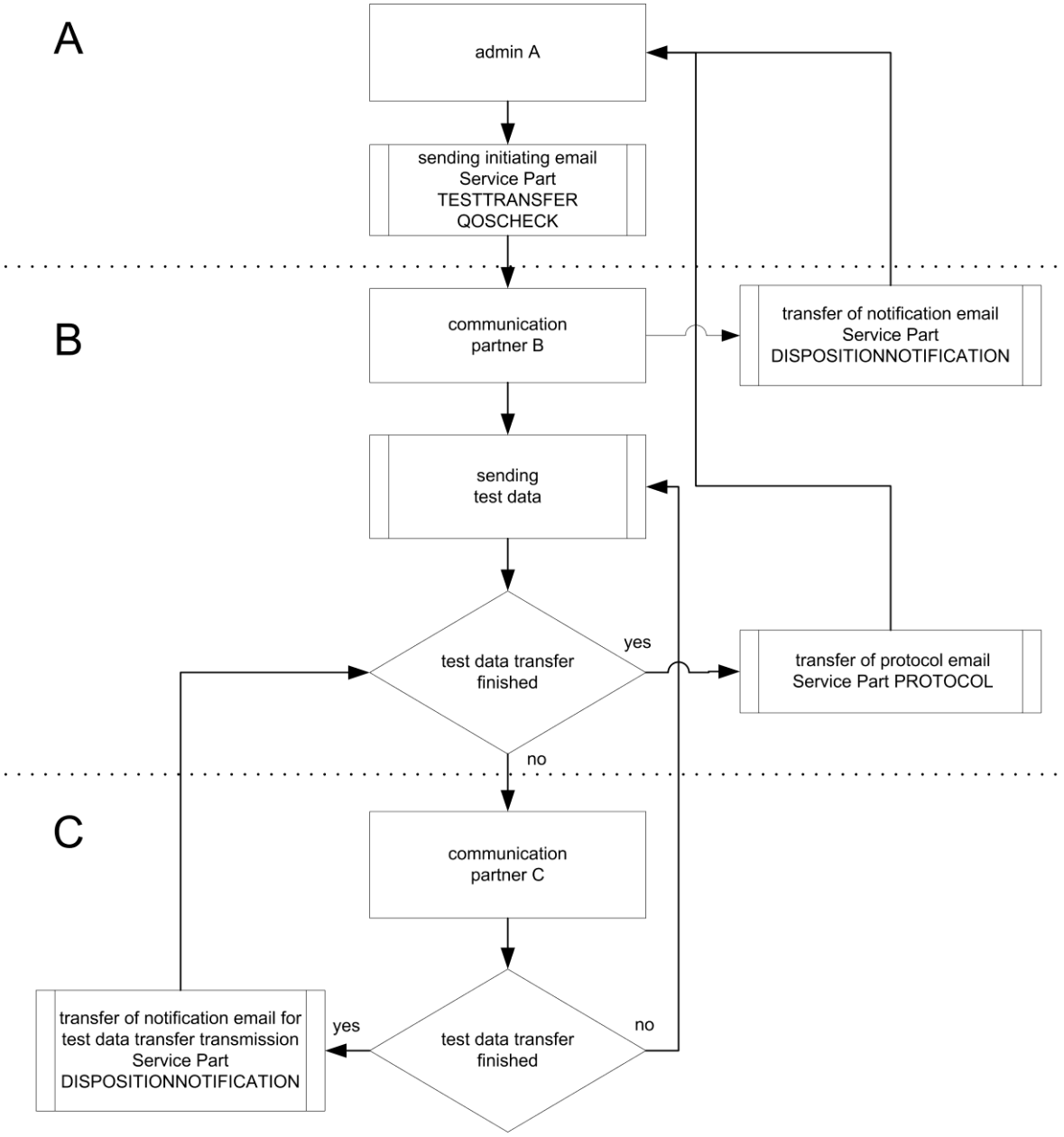

Fig. 64 – Sample process: constancy test by Service Part trigger e-mail

The following Fig. 65 – Example Service Part trigger shows an example of a trigger e-mail for the scenario described above (Fig. 65 – Example Service Part trigger e-mail).

```
From: admin_A@teleradiology.de
To: partner_B@teleradiologie.de
Subject: Request for testtransfer
```

```
Date: Fri, 14 Jun 2013 10:00:35 +0100
MIME-Version: 1.0
Message-Id: 83BF3401-0840-4268-83CD-610AF6259FD6@teleradiology.de
X-TELEMEDICINE-SERVICEPART: TESTTRANSFER
X-TELEMEDICINE-VERSION: 1.7.0
Disposition-Notification-To: admin_A@teleradiology.de
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="01BF27A0.9BE21980/teleradiology.de"

--01BF27A0.9BE21980/teleradiology.de
Content-Type: application/pgp-encrypted

Version: 1

--01BF27A0.9BE21980/teleradiology.de
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: PGP 8.0.1*

%Content-Disposition: attachment; filename="ServicePart.xml"
%Content-Type: text/xml; // application/xml is also valid but not recommended
%Content-Transfer-Encoding: quoted-printable
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="TESTTRANSFER" action="QOSCHECK" timestamp="YYYY-MM-DDThh:mm:ssZ">
%  <TestDataReceiver>
%   <EMailAddress>partner_C@homeoffice.de</EMailAddress>
%   <GPGKeyID>EA12FE67AA76CA08</GPGKeyID>
%  </TestDataReceiver>
%
%  <ProtocolReceiver>
%   <EMailAddress>admin_A@teleradiology.de </EMailAddress>
%   <GPGKeyID>5DE32FAFD4762A08</GPGKeyID>
%  </ProtocolReceiver>
%
%  <TestDataSetID>TESTDATASET_1</TestDataSetID>
%</ServicePart>

-----END PGP MESSAGE-----

--01BF27A0.9BE21980/teleradiology.de--
```

* The lines marked % are PGP/OpenPGP encrypted.

Fig. 65 – Example Service Part trigger e-mail (TESTTRANSFER QOSCHECK)

This sample e-mail (Fig. 65) should trigger the following actions:

1. On the side of the receiver, the defined data set (TESTDATASET_1) is sent to the e-mail address partner_C@homeoffice.de, encrypted with the key identified by the KeyID EA12FE67AA76CA08.
2. The protocol XML file (cf. Fig. 54 – XML structure Service Part PROTOCOL) is sent to the e-mail address admin_A@teleradiology.de. For encryption, the key with the KeyID 5DE32FAFD4762A08 is used.
3. A notification e-mail for the initial trigger e-mail (Fig. 65) should be sent to the address admin_A@teleradiology.de by partner_B@teleradiologie.de.

The Service Part PROTOCOL e-mail for this scenario is listed in the following example *Fig. 66 – Example Service Part PROTOCOL e-mail* (* The lines marked % are PGP/OpenPGP encrypted).

```
From: partner_B@teleradiologie.de
To: admin_A@teleradiology.de
Message-ID: <31175293.51336059080045@PC-123>
Subject: Mechanism type 3 notification e-mail
Date: Fri, 14 Jun 2013 09:14:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="-----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40

%Content-Type: multipart/mixed;
%boundary="-----=_Part_894967_718503476.1368448709295"
%Content-Disposition: attachment; filename="AnyName.xml"
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%
%------=_Part_894967_718503476.1368448709295
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="PROTOCOL" timestamp="2013-06-14T09:11:27Z">
% <TransmissionStatus>COMPLETED</TransmissionStatus>
% <TestDataSetID>TESTDATASET_1</TestDataSetID>
% <ErrorTimeOut>55554</ErrorTimeOut>
% <ObjectsSent>
%  <Count>2</Count>
% </ObjectsSent>
% <ObjectsReceivedConfirmed>
%  <Count>2</Count>
%  <Time>1256</Time>
%  <MailSize>1000</MailSize>
%  <ObjectSize>800</ObjectSize>
% </ObjectsReceivedConfirmed>
% <DataSender>
%  <EMailAddress>partner_B@teleradiologie.de</EMailAddress>
% </DataSender>
% <DataRecipient>
%  <EMailAddress>partner_C@homeoffice.de</EMailAddress>
% </DataRecipient>
% <ProtocolRecipient>
%  <EMailAddress>admin_A@teleradiology.de</EMailAddress>
% </ProtocolRecipient>
% <DatagramMail EMailMessageID="MAIL_x_0">
%  <ErrorID>1</ErrorID>
%  <EMailContentID>content_0</EMailContentID>
%  <StartDateTime>20130614111010</StartDateTime>
%  <NotifyDateTime>20130614111010</NotifyDateTime>
%  <MailSize>500</MailSize>
%  <ObjectSize>400</ObjectSize>
% </DatagramMail>
% <DatagramMail EMailMessageID="MAIL_x_1">
%  <EMailContentID>content_1</EMailContentID>
%  <StartDateTime>20130614111010</StartDateTime>
%  <NotifyDateTime>20130614111010</NotifyDateTime>
%  <MailSize>500</MailSize>
%  <ObjectSize>400</ObjectSize>
% </DatagramMail>
%</ServicePart>
------=_Part_894967_718503476.1368448709295--
```

Fig. 66 – Example Service Part PROTOCOL e-mail

19.2 Exchange of key data (set)

Service Parts: *KEYUPDATE SET*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 67 – Flowchart KEYUPDATE SET) shows how administrator A distributes his new OpenPGP key to administrator B. Therefore a Service Part KEYUPDATE SET is sent to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION.

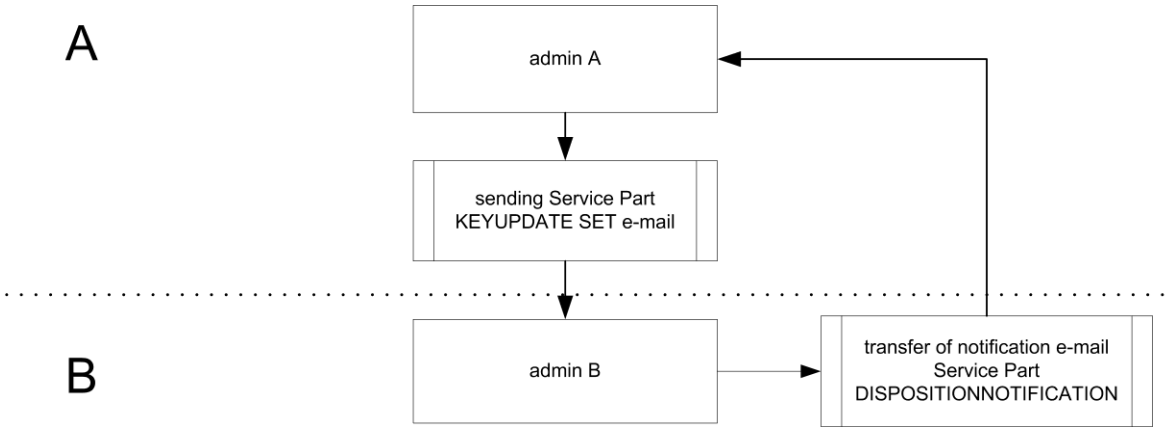

Fig. 67 – Flowchart KEYUPDATE SET

19.3 Exchange of key data (get)

Service Parts: *KEYUPDATE GET*
DISPOSITIONNOTIFICATION
KEYUPDATE SET

The following scenario (Fig. 68 – Flowchart KEYUPDATE GET) shows how administrator A updates his stock of OpenPGP encryption certificates from administrator B. Therefore, a Service Part KEYUPDATE GET is sent to administrator B (black arrows). The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION (black arrow) and a Service Part KEYUPDATE SET message which will be acknowledged by returning a Service Part DISPOSITIONNOTIFICATION e-mail (red arrows).

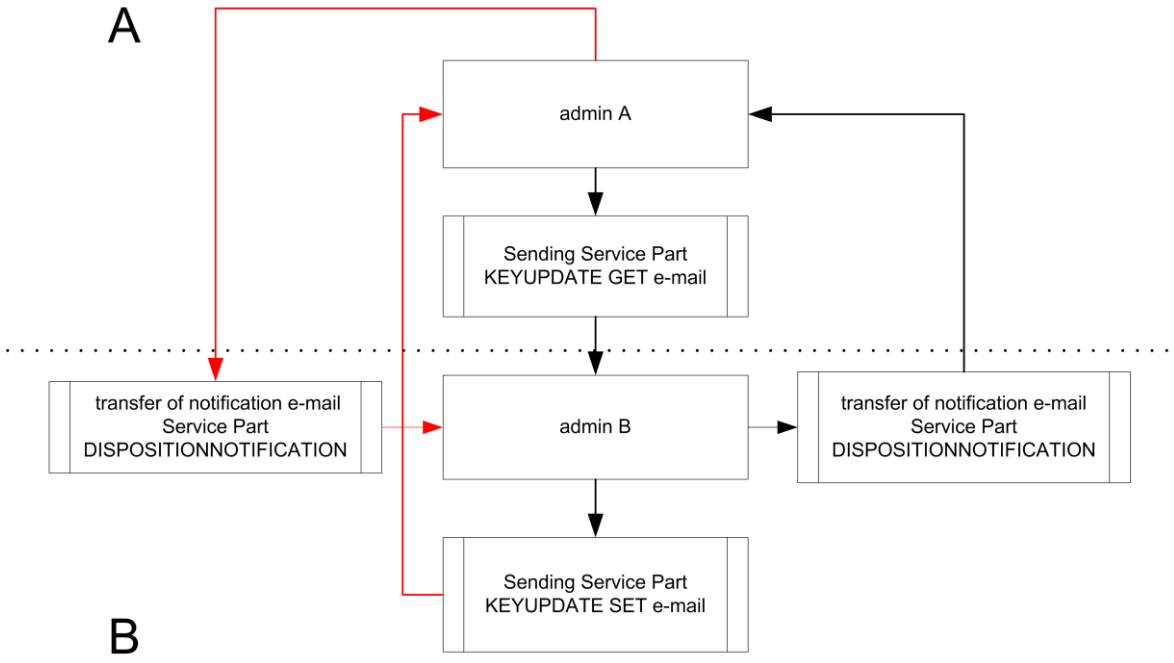

Fig. 68 – Flowchart KEYUPDATE GET

19.4 Exchange of key data (remove)

Service Parts: *KEYUPDATE REMOVE*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 69 – Flowchart KEYUPDATE REMOVE) shows how administrator A triggers the deletion of a specific key, defined by GPGKeyID, from the configuration of administrator B. Therefore, he sends a Service Part KEYUPDATE REMOVE to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION.

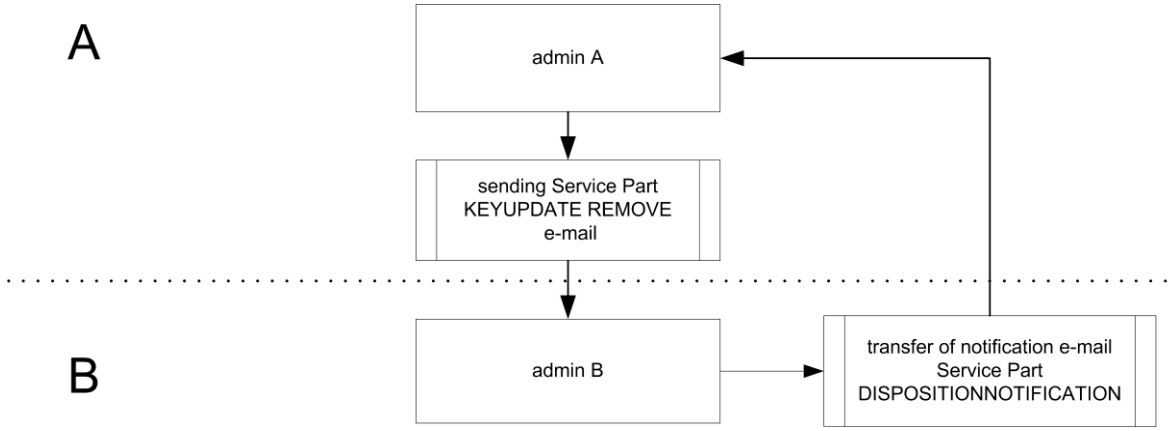

Fig. 69 – Flowchart KEYUPDATE REMOVE

19.5 Exchange of key data (clean)

Service Parts: *KEYUPDATE CLEAN*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 70 – Flowchart KEYUPDATE CLEAN) shows how administrator A triggers the deletion of all configured OpenPGP keys on the system of administrator B, apart from those he marked using the element KeepGPGKeyID. Therefore, he sends a Service Part KEYUPDATE CLEAN to administrator B. Administrator B acknowledges the reception of the Service Part e-mail by returning a Service Part DISPOSITIONNOTIFICATION.

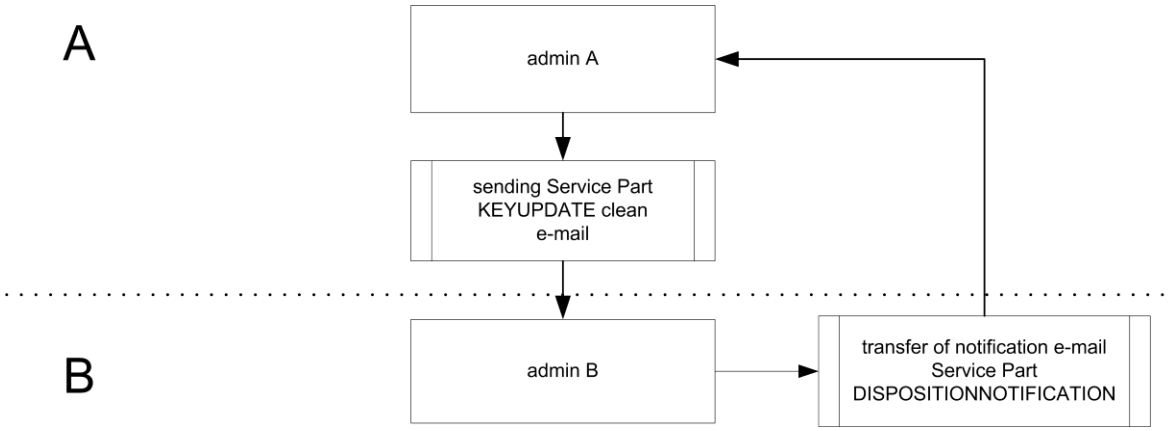

Fig. 70 – Flowchart KEYUPDATE CLEAN

19.6 Exchange of address data (set)

Service Parts: *ADDRESSUPDATE SET*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 71 – Flowchart ADDRESSUPDATE SET) shows how administrator A distributes new or updated address data to administrator B. Therefore, he sends a Service Part ADDRESSUPDATE SET to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION.

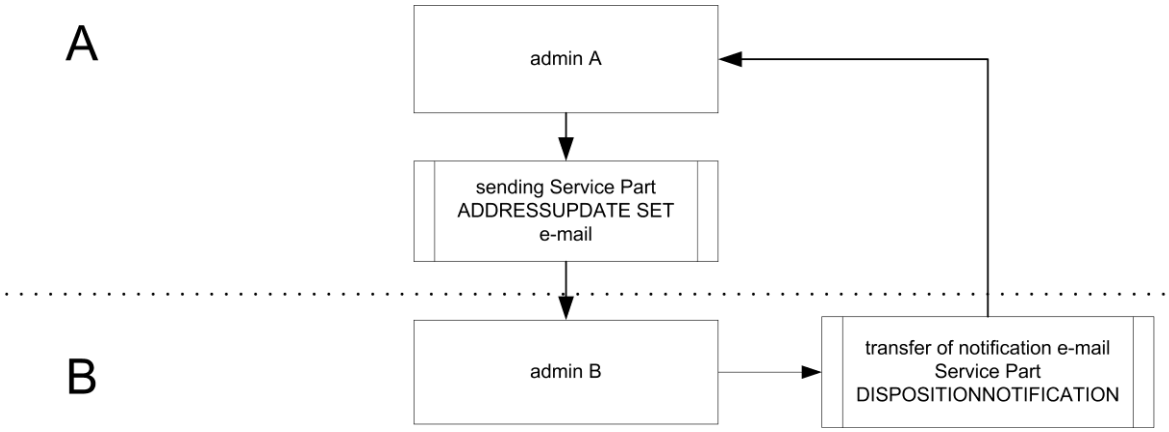

Fig. 71 – Flowchart ADDRESSUPDATE SET

19.7 Exchange of addressdata (get)

Service Parts: *ADDRESSUPDATE GET*
DISPOSITIONNOTIFICATION
ADDRESSUPDATE SET

The following scenario (Fig. 72 – Flowchart ADDRESSUPDATE GET) shows how administrator A updates specific address data, defined by ID, from the system of administrator B. Therefore, he sends a Service Part ADDRESSUPDATE GET to administrator B (black arrows). The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION (black arrow) and a Service Part ADDRESSUPDATE SET message for the specific ID which will be acknowledged from administrator A by returning a Service Part DISPOSITIONNOTIFICATION e-mail (red arrows) .

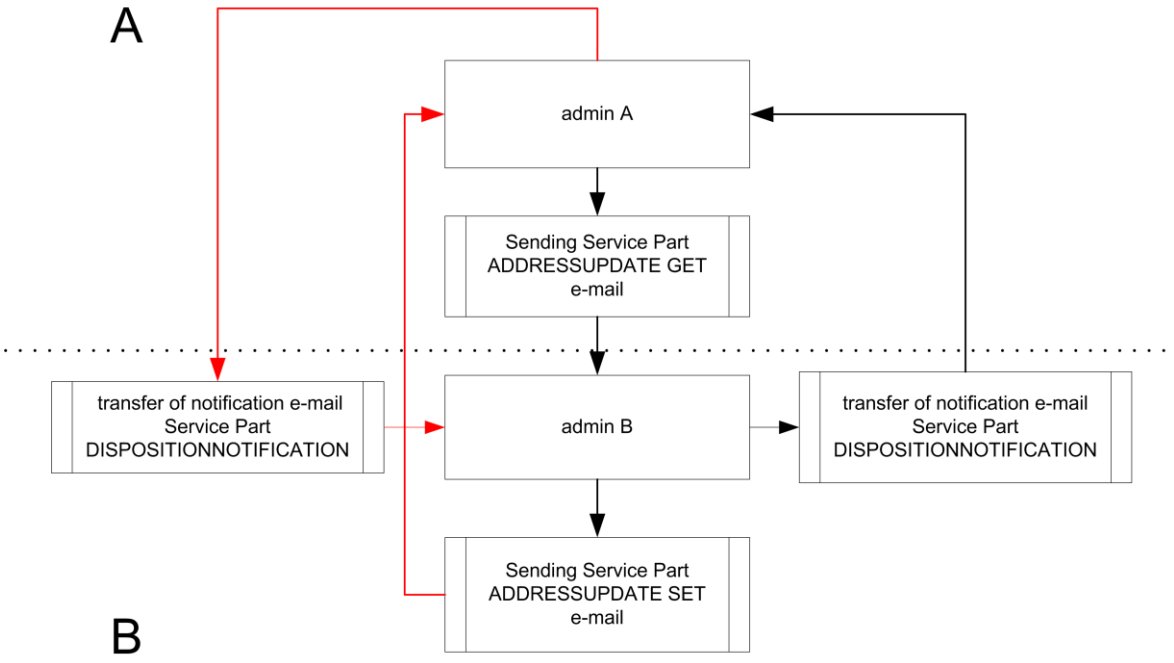

Fig. 72 – Flowchart ADDRESSUPDATE GET

19.8 Exchange of address data (remove)

Service Parts: *ADDRESSUPDATE REMOVE*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 73 – Flowchart ADDRESSUPDATE REMOVE) shows how administrator A triggers the deletion of address data, defined by ID, from the configuration of administrator B. Therefore, he sends a Service Part ADDRESSUPDATE REMOVE to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION.

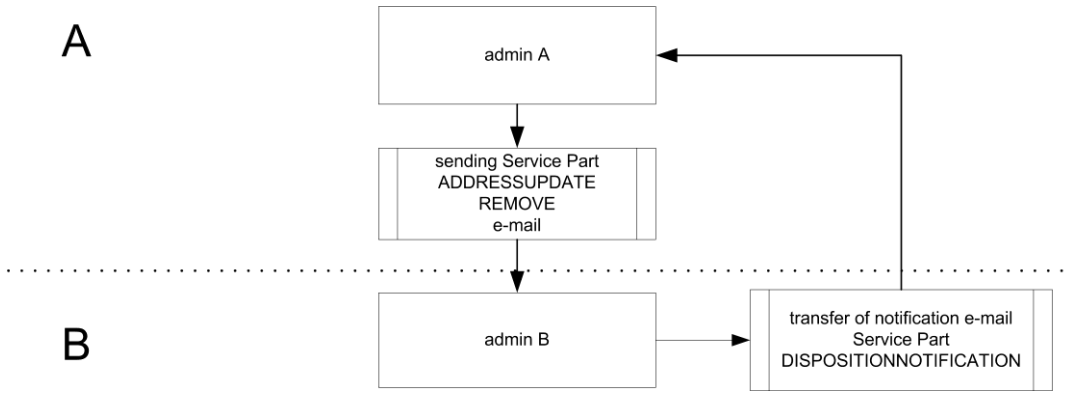

Fig. 73 – Flowchart ADDRESSUPDATE REMOVE

19.9 Exchange of address data (clean)

Service Parts: *ADDRESSUPDATE CLEAN*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 74 – Flowchart ADDRESSUPDATE CLEAN) shows how administrator A triggers the deletion of all configured connection address data on the system of administrator B, apart from those he marked using the element KeepConnectionID. Therefore, he sends a Service Part ADDRESSUPDATE CLEAN to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION.

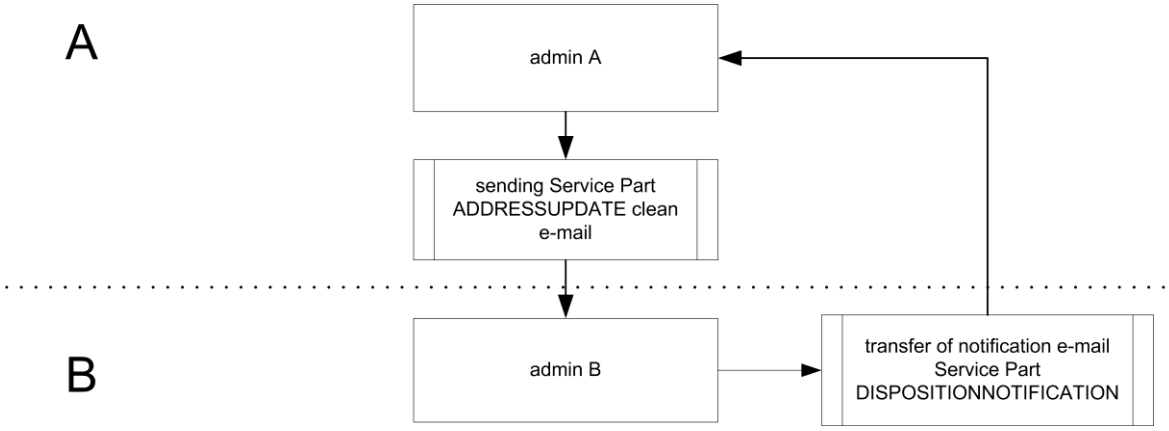

Fig. 74 – Flowchart ADDRESSUPDATE CLEAN

19.10 Exchange of connection data (get)

Used X-Tags: *X-TELEMEDICINE-SERVICEPART: CONNECTIONUPDATE*
DISPOSITIONNOTIFICATION

In order to update all data describing a connection (address data, encryption key data and contact data) this scenario shows how to update all these different data types by using only one Service Part.

The following scenario (Fig. 75 – Flowchart CONNECTIONDATA GET Single Service Part Scenario) shows how administrator A updates all connection data for a specific connection, defined by ID, retrieving this information from the system of administrator B. Therefore, he sends a Service Part CONNECTIONUPDATE GET to administrator B (black arrows). The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION (black arrow) and the Service Parts ADDRESSUPDATE SET, KEYUPDATE SET and CONTACTUPDATE SET. These responses can either be sent as three independent Service Part e-mails (Fig. 75 – Flowchart CONNECTIONDATA GET

Single Service Part Scenario; red, blue and yellow arrows) or one Service Part Multipart message (Fig. 76 – Flowchart CONNECTIONDATA GET Multi Service Part Scenario; red arrows) for the specific ID. The Service Parts will be acknowledged by administrator A by either returning three Service Part DISPOSITIONNOTIFICATION e-mails for each of the independently sent Service Part e-mails (Fig. 75 – Flowchart CONNECTIONDATA GET Single Service Part Scenario; red, blue and yellow arrows) or one Service Part DISPOSITIONNOTIFICATION in case of answering the Service Part Multipart message (Fig. 76 – Flowchart CONNECTIONDATA GET Multi Service Part Scenario; red arrows).

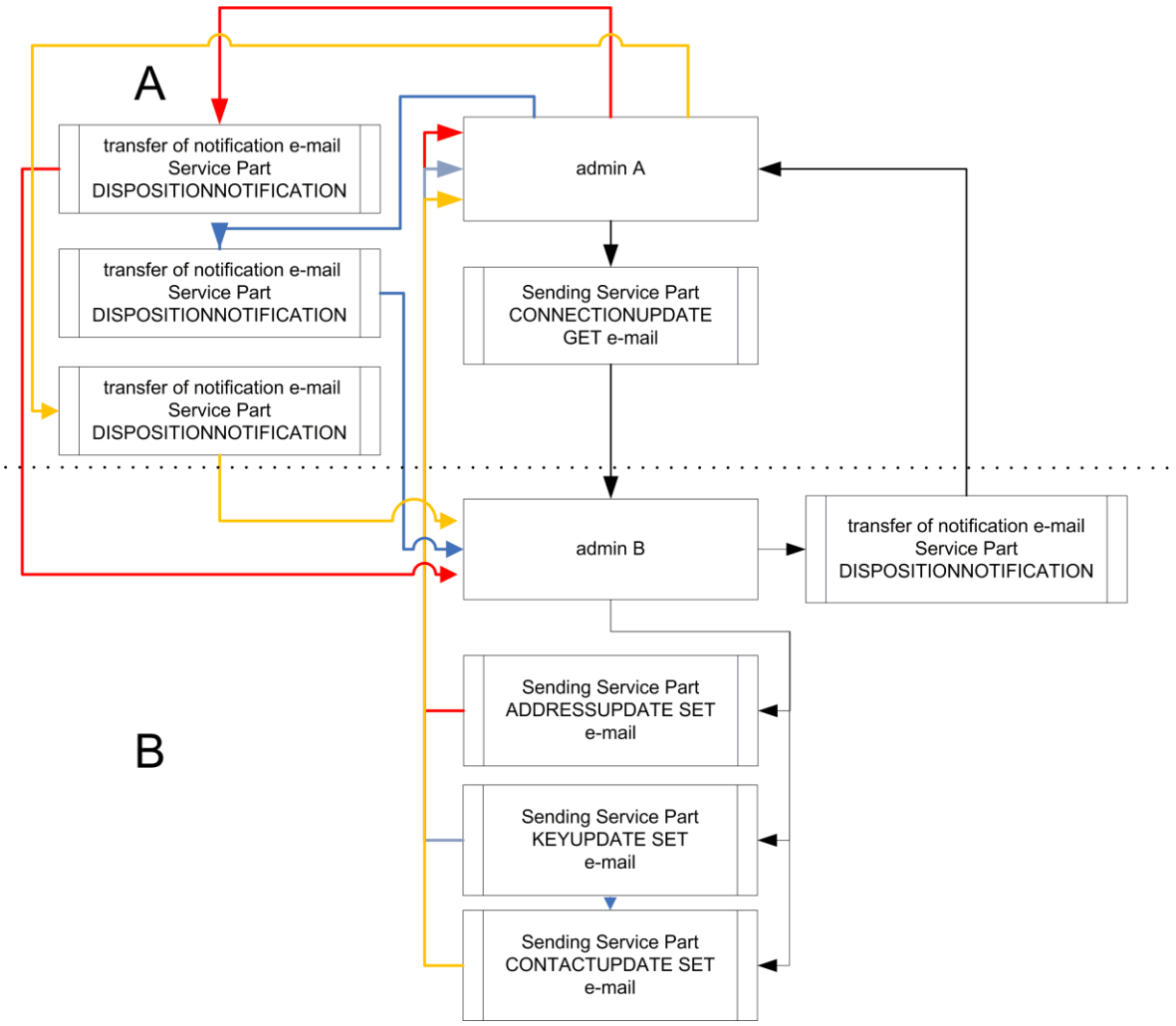

Fig. 75 – Flowchart CONNECTIONDATA GET Single Service Part Scenario

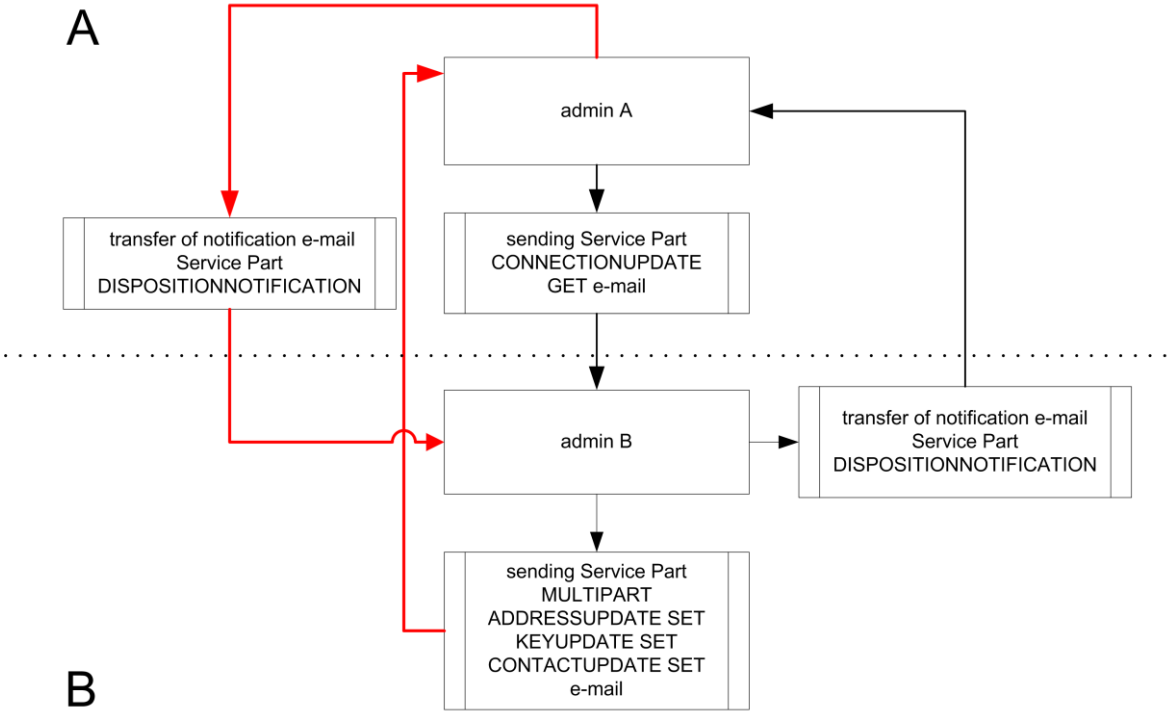

Fig. 76 – Flowchart CONNECTIONDATA GET Multi Service Part Scenario

19.11 Exchange of contact data (clean)

Service Parts: *CONTACTUPDATE CLEAN*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 77 – Flowchart CONTACTUPDATE CLEAN) shows how administrator A triggers the deletion of all configured contact data at administrator B apart from these he marked using the element KeepContactID. Therefore he sends a Service Part CONTACTUPDATE CLEAN to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION e-mail.

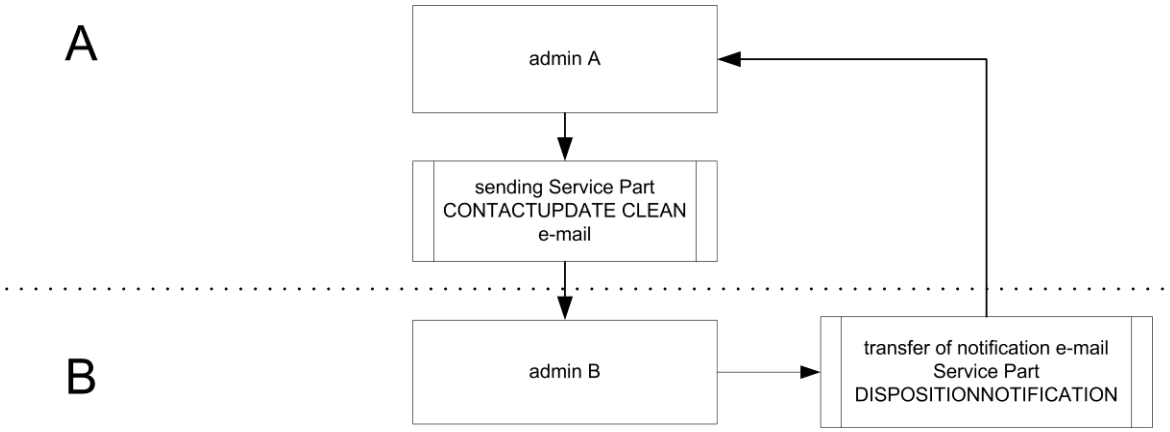

Fig. 77 – Flowchart CONTACTUPDATE CLEAN

19.12 Exchange of contact data (get)

Service Parts: *CONTACTUPDATE GET*
CONTACTUPDATE SET
DISPOSITIONNOTIFICATION

The following scenario (Fig. 78 – Flowchart CONTACTUPDATE GET) shows how administrator A updates specific contact data from administrator B. Therefore he sends a Service Part CONTACTUPDATE GET to administrator B (black arrows). The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION and a Service Part CONTACTUPDATE SET message (red arrow) which will be acknowledged from administrator A by returning a Service Part DISPOSITIONNOTIFICATION e-mail (red arrows).

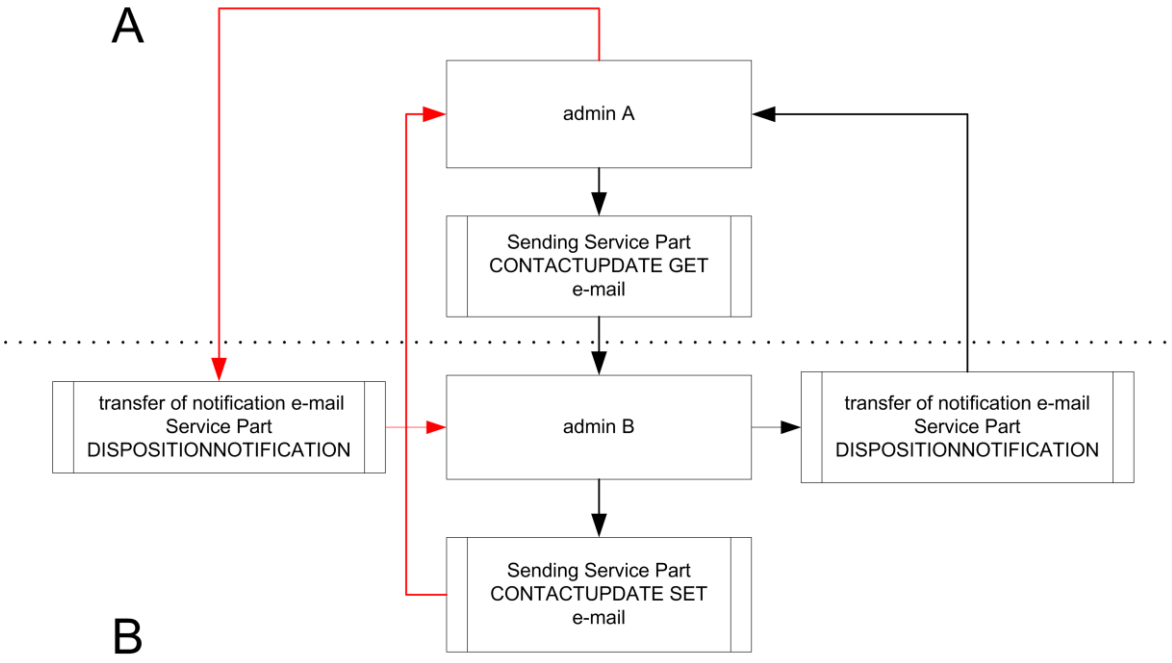

Fig. 78 – Flowchart CONTACTUPDATE GET

19.13 Exchange of contact data (remove)

Service Parts: *CONTACTUPDATE REMOVE*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 79 – Flowchart CONTACTUPDATE REMOVE) shows how administrator A triggers the deletion of contact data, defined by ContactID, from the configuration of administrator B. Therefore, he sends a Service Part CONTACTUPDATE REMOVE to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION e-mail.

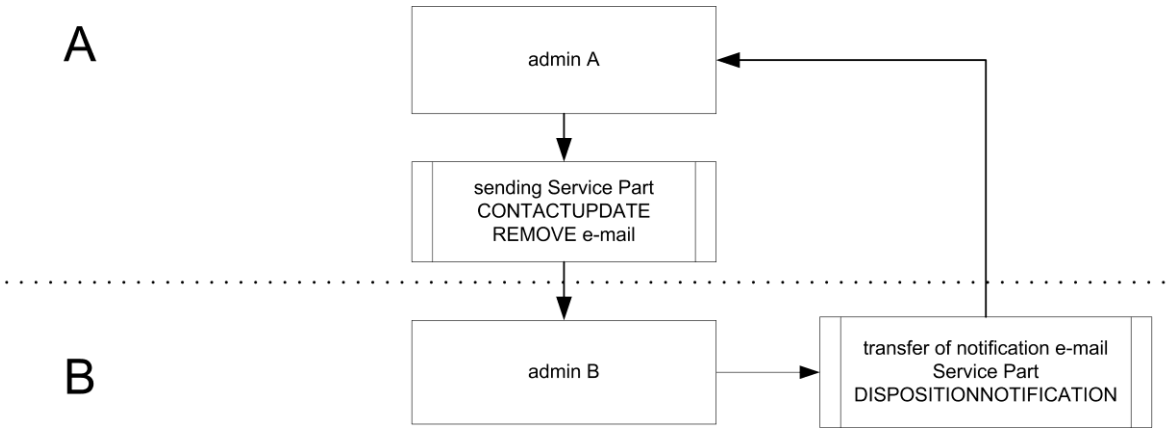

Fig. 79 – Flowchart CONTACTUPDATE REMOVE

19.14 Exchange of contact data (set)

Service Parts: *CONTACTUPDATE SET*
DISPOSITIONNOTIFICATION

The following scenario (Fig. 80 – Flowchart CONTACTUPDATE SET) shows how administrator A distributes new or updated contact data to administrator B. Therefore, he sends a Service Part CONTACTUPDATE SET to administrator B. The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION e-mail.

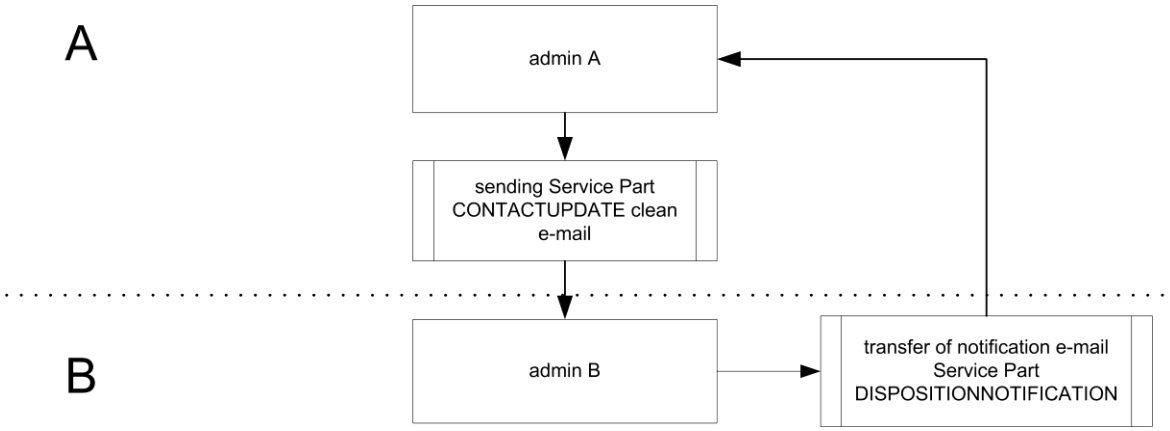

Fig. 80 – Flowchart CONTACTUPDATE SET

19.15 Query and retrieve connection data

Service Parts: *QUERY FIND*
QUERY RESULT
DISPOSITIONNOTIFICATION

The following scenario (Fig. 81 – Flowchart QUERY FIND and RETRIEVE) shows how administrator A retrieves any connection data from administrator B. Therefore, he sends a Service Part QUERY FIND to administrator B (black arrows). The reception of the Service Part e-mail is acknowledged by returning a Service Part DISPOSITIONNOTIFICATION (black arrow) and a Service Part QUERY RESULT message (red arrow) containing the data specified by the element Filter and ResultSet which will be acknowledged from administrator A by returning a Service Part DISPOSITIONNOTIFICATION e-mail (red arrows).

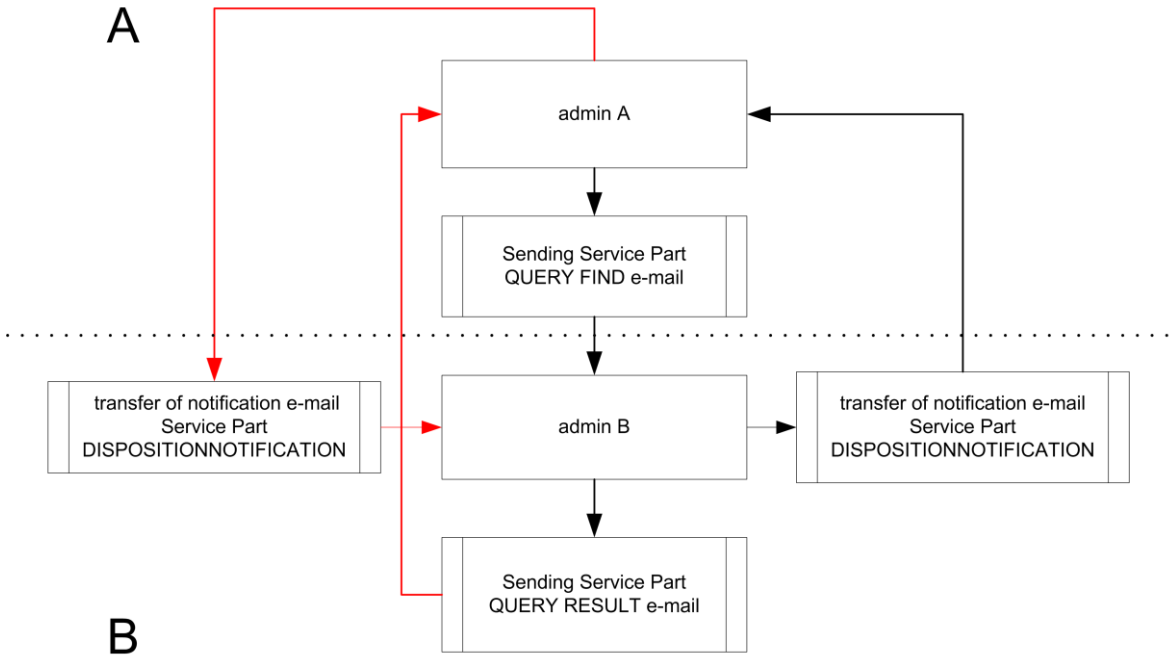

Fig. 81 – Flowchart QUERY FIND and RETRIEVE

20 Generating identification numbers

20.1 Advice

Please note that in the context of the telemedicine, identification numbers (e.g. *X-TELEMEDICINE-STUDYID*, *X-TELEMEDICINE-SETID,…) must not* allow any inferences regarding patients' identity.

20.2 DICOM UID

The DICOM-Root-UIDs required for the generation of valid DICOM UIDs can be obtained free of charge via Medical Connections (http://www.medicalconnections.co.uk/Free_UID) or are commercially available in the official institution responsible for your country.

21 Further applicable documents

21.1 RFC

RFC 4122	A Universally Unique IDentifier (UUID) URN Namespace
RFC2392	Content-ID and Message-ID Uniform Resource Locators
RFC 3339	Date and Time on the Internet: Timestamps
RFC 3240	Digital Imaging and Communications in Medicine (DICOM) - Application/dicom MIME Sub-type Registration
RFC2195	IMAP/POP AUTHorize Extension for Simple Challenge/Response
RFC3510	INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1
RFC3798	Message Disposition Notification
RFC3156	MIME Security with OpenPGP
RFC2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
RFC2046	Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
RFC4880	OpenPGP Message Format
RFC5034	POP3 AUTHentication command
RFC1939/ STD0053	Post Office Protocol - Version 3
RFC1847	Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted
RFC5321	Simple Mail Transfer Protocol
RFC1846	SMTP 521 Reply Code
RFC6152	SMTP Service Extension for 8bit-MIMEtransport
RFC4954	SMTP Service Extension for Authentication
RFC2034	SMTP Service Extension for Returning Enhanced Error Codes
RFC3207	SMTP Service Extension for Secure SMTP over Transport Layer Security
RFC3030	SMTP Service Extensions for Transmission of Large and Binary MIME Messages
RFC6522	The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages
RFC3206	The SYS and AUTH POP Response Codes

RFC2595	Using TLS with IMAP, POP3 and ACAP
RFC 3023	XML Media Types

21.2 DICOM Standard

- DICOM Standard, Suppl. 54 - DICOM MIME Content-Type

21.3 German Regulations

- Verordnung über den Schutz vor Schäden durch Röntgenstrahlen („Röntgenverordnung“, *German X-ray Ordinance*) – Revised Version Announced 30 April 2003 I 604
- Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz, SigG, *Law on the General Requirements for Electronic Signatures*) – Status: Modified by Art. 1 G, 4 January 2005 I 2
- Verordnung zur elektronischen Signatur (Signaturverordnung, *Electronic Signature Ordinance*) – Status: Modified by Art. 2 G, 4 January 2005 I 2

22 Appendix Examples

22.1 Multipart Service Part Message using Notify Mechanism 3

```
From: from@test.test
To: to@test.test
Message-ID: <31175293.51336059080045@PC-CVS>
Subject: Multipart Service Part Example
Date: Fri, 4 May 2013 07:40:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: MULTIPART
Disposition-Notification-To: from@test.test

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40

%Content-Type: multipart/mixed; boundary="----=_Part_4_7272378.1336117204881"
%
%------=_Part_4_7272378.1336117204881
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: 7bit
%Content-ID: KEYUPDATE_10_13e9de5c461_4ec769a13e580e63
%X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-KEYID= <PGP keyid 1>,<PGP keyid 2>,...
%X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-TO= <e-mail addr1>,<e-mail addr2>,...
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart action="SET" name="KEYUPDATE" timestamp="2012-05-04T07:40:04Z">
% <PublicKeyASCIIData>
% -----BEGIN PGP PUBLIC KEY BLOCK-----
% Version: BCPG v1.40
%
% 0Wh7seClfszkKVpnfiVY+thZmFw2r/odGXPLl88dXK7mzF8bFv1DmtrMG2mjzyk7MXX3eEZ
% kRD+Uj5B/DKuLbOIMwphEvu8+RGmEMPiB+wDMrrZdZwMzeb6JDS69Rs2sGKSdr4bOU3STPc
% geokWVS1A0qUOoljbkuWBrZPLuU21D3MJZVZcMeXW5GoOXkwMMfkUJzZ2Jq2QljT1hnV/lJ
% inNPLSb9GCDD2xqKJMj5pSS0+gGg75UCtUPr6RXtSpkB+DHW4lsN6C9eaZvnAmoqY4eSJVA
% ncNDKevCqb0LQfuFf2Mtv8VHhnfWr11jN7uFTmrxpsD7apEjwe12C5BQlwQcUAJ3DVHYa4e
% -----END PGP PUBLIC KEY BLOCK-----
% </PublicKeyASCIIData>
%</ServicePart>
%------=_Part_4_7272378.1336117204881
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: 7bit
%Content-ID: ADDRESSUPDATE_ca9a8e13-4e1f-4745-86b8-3bca6e6a507d
%X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-KEYID= <PGP keyid 1>,<PGP keyid 2>,...
%X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-TO= <e-mail addr1>,<e-mail addr2>,...
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart action="SET" name="ADDRESSUPDATE" timestamp="2012-05-04T07:40:04Z">
% <Connection>
%  <Mailserver>9AD1388C55FBDEC101745FD2</Mailserver>
%  <Port>995</Port>
%  <ID>2de73e58-9691-4675-8760-50cf0906dd8f</ID>
%  <DisplayConnectionName>3203DB0F9F485B1896DE53DE</DisplayConnectionName>
%  <EMailAddress>22019B72E8377E59256BDDC8@test.net</EMailAddress>
%  <GPGKeyID>22019B72E8377E59256BDDC8</GPGKeyID>
% </Connection>
%</ServicePart>
%------=_Part_4_7272378.1336117204881
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%Content-ID: CONTACTUPDATE_11_13e9de5c464_189af4ed5e6784f3
%
%<?xml version="1.0" encoding="UTF-8"?>
```

```
%<ServicePart action="SET" name="CONTACTUPDATE" timestamp="2013-05-15T09:38:36Z">
% <ContactID>daeca6ec-2bdd-493c-a4ff-2b65606eea47</ContactID>
% <ConnectionID>0ba425d3-3230-4591-9a8e-a42ce537cdda</ConnectionID>
% <RelatedID>d7d675c0-288d-4098-9849-318bce3594c9</RelatedID>
% <Name>Klinik B</Name>
% <Unit>ORGANISATION</Unit>
% <Comment/>
% <ContactInformation order="1">
%  <Fax/>
%  <Mobile/>
%  <Pager/>
   <Website/>
% </ContactInformation>
% <Address order="1">
%  <Street_1/>
%  <Street_2/>
%  <City>Dortmund</City>
%  <ZipCode/>
%  <State/>
%  <Country/>
% </Address>
%</ServicePart>
%------=_Part_4_7272378.1336117204881
```

22.2 Example Service Part Notification e-mail in response to a multipart Service Part mail

```
From: from@test.test
To: to@test.test
Message-ID: <31175293.51336059080045@PC-123>
Subject: Mechanism type 3 notification e-mail
Date: Sun, 13 May 2013 12:39:35 +0000
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
boundary="----=_Part_5_10697954.1336059080045"
X-TELEMEDICINE-SERVICEPART: DISPOSITIONNOTIFICATION
X-TELEMEDICINE-VERSION:1.7.0

------=_Part_5_10697954.1336059080045
Content-Type: application/pgp-encrypted
Content-Transfer-Encoding: 7bit

Version: 1
------=_Part_5_10697954.1336059080045
Content-Type: application/octet-stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: BCPG v1.40

%Content-Type: multipart/mixed;
%boundary="----=_Part_894967_718503476.1368448709295"
%
%------=_Part_894967_718503476.1368448709295
%Content-Type: text/xml; charset=UTF-8
%Content-Transfer-Encoding: quoted-printable
%
%<?xml version="1.0" encoding="UTF-8"?>
%<ServicePart name="DISPOSITIONNOTIFICATION" timestamp="2013-05-13T12:38:29Z">
%    <MessageID>31175293.51336059080045@PC-CVS</MessageID>
%    <Notification>
%        <ContentID>ADDRESSUPDATE_ca9a8e13-4e1f-4745-86b8-3bca6e6a507d</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%    <Notification>
%        <ContentID>KEYUPDATE_10_13e9de5c461_4ec769a13e580e63</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
```

```
%    <Notification>
%        <ContentID>CONTACTUPDATE_11_13e9de5c464_189af4ed5e6784f3</ContentID>
%        <DispositionField>displayed</DispositionField>
%    </Notification>
%</ServicePart>
------=_Part_894967_718503476.1368448709295—
```

23 Appendix Overview of all X-TELEMEDICINE Tags

X-tag	Description	Ver.
X-TELEMEDICINE-DISPOSITION-NOTIFCATION-KEYID	Requests encrypted X-Telemedicine notifications for this gpg keyid	1.5-1.6.1
X-TELEMEDICINE-DISPOSITION-NOTIFICATION-TO	Requests x-telemedicine notifications for this e-mail address	1.5-1.6.1
X-TELEMEDICINE-ORIGINAL-CONTENT-ID	Original content ID for encrypted confirmation e-mails.	1.5 - 1.6.1
X-TELEMEDICINE-SERVICEPART:<ACTION>	Identifies Service Part operations e.g. KEYUPDATE.	1.6+
X-TELEMEDICINE-SERVICEPART:DISPOSITIONNOTIFICATION	Marks the e-mail as a notification Service Part e-mail.	1.7+
X-TELEMEDICINE-SERVICEPART:MULTIPART	Service Part e-mails containing more than one Service Part.	1.7+
X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-KEYID:	X-Tag for providing the encryption key IDs (long version) that must be used to encrypt the notification e-mail content (mechanism 3).	1.7+
X-TELEMEDICINE-SERVICEPART-DISPOSITION-NOTIFICATION-TO:	X-Tag for providing the notification e-mail recipient addresses (mechanism 3).	1.7+
X-TELEMEDICINE-SETID	For grouping of associated e-mails to sets.	1.5+
X-TELEMEDICINE-SETPART	Identifies an e-mail belonging to a set.	1.5+
X-TELEMEDICINE-SETTOTAL	Identifies the last e-mail belonging to a set (total number of all parts).	1.5+
X-TELEMEDICINE-STUDYID	For mapping NON-DICOM data against DICOM studies.	1.1+
X-TELEMEDICINE-VERSION:x.y.z	Version of the Standard Recommendation main version=x, sub version=y, revision=z	1.7+

24 Appendix Error Codes

Code	Meaning
Undefined	
0	vendor specific errors
Mail	
1	mail-error
1.1	mail-receipt-error
1.1.1	mail-receipt-failed
1.1.2	mail-receipt-was-read-before
1.2	mail-syntax-error
1.2.1	mail-syntax-header-error
1.2.1.1	mail-syntax-header-contentid-error
1.2.1.1.1	mail-syntax-header-contentid-missing
1.2.1.2	mail-syntax-header-dispo_to-error
1.2.1.2.1	mail-syntax-header-dispo_to-missing
1.2.1.3	mail-syntax-header-contenttype
1.2.1.3.1	mail-syntax-header-contenttype-missing
1.2.2	mail-syntax-body-error
1.2.2.1	mail-syntax-body-empty
1.2.2.2	mail-syntax-body-missing
1.3	mail-attachment-error
1.3.1	mail-attachment-corrupt
1.4	mail-mimetype-error
1.4.1	mail-mimetype-not-processed
1.4.2	mail-mimetype-not-supported
1.5	mail-security-error
1.5.1	mail-security-signature-error
1.5.1.1	mail-security-signature-missing
1.5.2	mail-security-encryption-error
1.5.2.1	mail-security-encryption-missing
1.6	mail-message/partial-error
1.6.1	mail-message/partial-part-error
1.6.1.1	mail-message/partial-part-missing
1.6.1.2	mail-message/partial-part-twice
1.6.1.3	mail-message/partial-part-header-error
1.6.1.3.1	mail-message/partial-part-header-id-error
1.6.1.3.1.1	mail-message/partial-part-header-id-missing
1.6.1.3.2	mail-message/partial-part-header-number-error
1.6.1.3.2.1	mail-message/partial-part-header-number-missing
1.6.1.3.3	mail-message/partial-part-header-total-error
1.6.1.3.3.1	mail-message/partial-part-header-total–missing
OpenPGP	
2	gpg-error
2.1	gpg-signature-error

2.1.1	gpg-signature-bad
2.1.2	gpg-signature-expired
2.2	gpg-key-error
2.2.1	gpg-key-expired
2.2.1.1	gpg-key-expired-sender
2.2.1.2	gpg-key-expired-receiver
2.2.2	gpg-key-revoked
2.2.2.1	gpg-key-revoked-sender
2.2.2.2	gpg-key-revoked-receiver
2.2.3	gpg-key-trust-error
2.2.3.1	gpg-key-trust-undefined
2.2.3.2	gpg-key-trust-never
2.2.3.3	gpg-key-trust-marginal
2.2.4	gpg-key-missing-error
2.2.4.1	gpg-key-missing-public
2.2.4.2	gpg-key-missing-private
2.2.5	gpg-key-signature-error
2.2.5.1	gpg-key-signature-expired
2.2.5.2	gpg-key-signature-revoked
2.3	gpg-passphrase-error
2.3.1	gpg-passphrase-bad
2.3.2	gpg-passphrase-missing
2.4	gpg-decryption-error
2.4.1	gpg-decryption–failed

Application

3	application-error
3.1	application-extern-error
3.2	application-intern-error
3.2.1	application-intern-attachment-error
3.2.1.1	application-intern-attachment-not-processed
3.2.2	application-intern-mimetype-error
3.2.2.1	application-intern-mimetype-unknown
3.2.2.2	application-intern-mimetype-not-processed
3.3	application-permission-error

XTelemedicine

4	x-telemedicine-error
4.1	x-telemedicine-studyid-error
4.1.1	x-telemedicine-studyid-missing-for-nondicom
4.1.2	x-telemedicine-studyid-not-allowed-for-dicom
4.2	x-telemedicine-set-tag-error
4.2.1	x-telemedicine-set-tag-content-differs
4.2.2	x-telemedicine-set-tag-intern-error
4.2.2.1	x-telemedicine-set-tag-intern-missing
4.2.2.2	x-telemedicine-set-tag-intern-id-error
4.2.2.2.1	x-telemedicine-set-tag-intern-id-missing
4.2.2.3	x-telemedicine-set-tag-intern-part-error
4.2.2.3.1	x-telemedicine-set-tag-intern-part-missing
4.2.2.4	x-telemedicine-set-tag-intern-total

4.2.2.4.1	x-telemedicine-set-tag-intern-total-missing
4.2.3	x-telemedicine-set-tag-extern-error
4.2.3.1	x-telemedicine-set-tag-extern-missing
4.2.3.2	x-telemedicine-set-tag-extern-differs
4.2.3.3	x-telemedicine-set-tag-extern-id-error
4.2.3.3.1	x-telemedicine-set-tag-extern-id-missing
4.2.3.3.2	x-telemedicine-set-tag-extern-id-differs
4.2.3.4	x-telemedicine-set-tag-extern-part-error
4.2.3.4.1	x-telemedicine-set-tag-extern-part-missing
4.2.3.4.2	x-telemedicine-set-tag-extern-part-differs
4.2.3.5	x-telemedicine-set-tag-extern-total-error
4.2.3.5.1	x-telemedicine-set-tag-extern-total-missing
4.2.3.5.2	x-telemedicine-set-tag-extern-total-differs
4.3	x-telemedicine-disposition-notification-tag-error
4.3.1	x-telemedicine-disposition-notification-tag-keyid-error
4.3.1.1	x-telemedicine-disposition-notification-tag-keyid-missing
4.3.2	x-telemedicine-disposition-notification-tag-to-error
4.4	x-telemedicine-contentid-error
4.4.1	x-telemedicine-contentid-missing

ServiceParts

5	servicepart-error
5.1	servicepart-protocol-error
5.1.1	servicepart-protocol-creation-error
5.2	servicepart-testtransfer-error
5.2.1	servicepart-testtransfer-testdataset-not-found
5.2.2	servicepart-testtransfer-testimages-not-found
5.3	servicepart-keyupdate-error
5.3.1	servicepart-keyupdate-addkey-error
5.3.2	servicepart-keyupdate-updatekey-error
5.3.3	servicepart-keyupdate-removekey-error
5.4	servicepart-addressupdate-error
5.4.1	servicepart-addressupdate-addaddress-error
5.4.2	servicepart-addressupdate-updateaddress-error
5.4.3	servicepart-addressupdate-removeaddress-error
5.5	servicepart-unsupported

Each error code can be extended by ".0" (user defined / application specific).

Example: 2.1.0.2 (potential meaning: no public key available, no GPG installed)

Proposals for further codes can be submitted using the website:
http://tele-x-standard.de

25 Appendix Test Data Set IDs

Requirements:

1. The test data set ID must consist of max. 64 alphanumeric characters.
2. Any number of test diagram data sets can be defined.
3. It is mandatory to implement the defined IDs as well as the possibility to add any further test diagram data sets.

Test Data Set ID	Descriptions
TESTDATASET_1	Function test data set (for the daily function test in accordance with DIN 6868-159)
TESTDATASET_2	Largest test diagram data set or equivalent (for the monthly constancy test according to DIN 6868-159)
TESTDATASET_CT_HEAD	Test data set CT head
TESTDATASET_CT_NECK	Test data set CT neck
TESTDATASET_CT_THORAX	Test data set CT thorax
TESTDATASET_CT_ABDOMEN	Test data set CT abdomen
TESTDATASET_CT_UPPER_EXTREMITY	Test data set CT upper extremities
TESTDATASET_CT_LOWER_EXTREMITY	Test data set CT lower extremities
TESTDATASET_CT_JOINTS	Test data set CT joints
TESTDATASET_CT_HAND	Test data set CT hand
TESTDATASET_CT_FOOT	Test data set CT foot
TESTDATASET_CT_FULLBODY_TRAUMA	Test data set CT full body/trauma
TESTDATASET_CT_ANGIO_ABDOMEN	Test data set CTA abdomen
TESTDATASET_CT_ANGIO_EXTREMITIES	Test data set CTA extremities
TESTDATASET_CR_THORAX	Test data set conventional X-ray images thorax
TESTDATASET_CR_ABDOMEN	Test data set conventional X-ray images abdomen
TESTDATASET_CR_EXTREMITIES	Test data set conventional X-ray images extremities

26 Glossary

Definition	Description
Communication path	This describes the connection between two communication partners with one sender and one receiver.
Connection data	This term describes all data defined by the XML structures of the Service Parts ADDRESSUPDATE, KEYUPDATE and CONNECTIONUPDATE.
DICOM E-MAIL	DICOM E-MAIL is the enhancement of the DICOM MIME type (DICOM Supl. 54 / DICOM Base Standard PS 3.11 and 3.12 / RFC 3240) by the Working Group on Information Technology of the German Radiology Society.
OpenPGP key ID	Each OpenPGP Key possesses a fingerprint, which is a unique identifier consisting of 40 hex digits. The key ID is represented by either the last 16 digits (long version) or the last 8 digits (short version). As of version 1.7 key ID always refers to the long version.