Test protocol for receiving a

CareCommunication

18-01-24

The test protocol relates to the following standard:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of the standard ENG** | **Name of the standard DK** | **Version** | **Type** |
| Standard: CareCommunication | Korrespondancemeddelelse | 3.0.X | HL7 FHIR |

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | | | |
| **Version** | **Initials** | **Date** | **Description** |
| 2.1.0 | KML/KRC/TMS/OVI | 31-03-2023 | First release |
| 3.0.0 | TMS/KRC | 18-01-2024 | Update of test protocol in accordance with release 3.0 of the documentation |

Content

[1 Introduction 4](#_Toc156468730)

[1.1 Purpose 4](#_Toc156468731)

[1.2 Prerequisites for live test 4](#_Toc156468732)

[1.3 Documentation of self-test 5](#_Toc156468733)

[1.4 Background material 6](#_Toc156468734)

[1.5 Test examples and test persons 7](#_Toc156468735)

[1.6 Test tools 7](#_Toc156468736)

[1.7 Test result 8](#_Toc156468737)

[2 Information about vendor, system under test (SUT) and test result information 9](#_Toc156468738)

[2.1 Information about the vendor 9](#_Toc156468739)

[2.2 Information about the system under test (SUT) 9](#_Toc156468740)

[2.3 Information about test results 9](#_Toc156468741)

[3 The test 10](#_Toc156468742)

[3.1. Documentation of the test 11](#_Toc156468743)

[3.2. Test of TouchStone test scripts 12](#_Toc156468744)

[3.3. Test of requirements to content and flow/workflows 13](#_Toc156468745)

[3.4. Test of general technical requirements 23](#_Toc156468746)

# Introduction

This is a test protocol for receiving a CareCommunication (DK: Korrespondancemeddelelse).

All documentation concerning CareCommunication and Governance (see [Background material](#_Baggrundsmaterialer)) will be the topic of testing, and the test protocol will be continuously updated to reflect the requirements in the best way possible.

Versioning of the test protocol will follow the major and minor versions of the standard but may have a patch version that is different from the standard’s patch version.

The test protocol will be available both in Danish and English. In case of any discrepancies between the two versions, the Danish version applies.

**As regards to sending of Acknowledgements**: To be approved, the system under test (SUT) must be approved for receiving the FHIR Acknowledgement (DK: Kvittering). This is described in the [test protocol for sending Acknowledgements](https://medcomdk.github.io/dk-medcom-acknowledgement/" \l "2-test-and-certification).

## Purpose

The test protocol forms the basis for the tests, which must ensure that SUT complies with the established rules and requirements for the standard. The test protocol also forms the basis for the self-test that vendors carry out prior to a live test.

## Prerequisites for live test

The following prerequisites must be met prior to the live test:

1. The vendor has read the following standard documentation, including:
   * [Clinical guidelines](#_Baggrundsmaterialer_1)
   * [Use cases](#_Baggrundsmaterialer_1)
   * [Implementation Guide](#_Baggrundsmaterialer_1)
   * [Governance](#_Baggrundsmaterialer_1)
   * And other relevant materials, cf. [background material](#_Baggrundsmaterialer_1).
2. The vendor has performed self-test, approved by MedCom
3. The vendor has created relevant test persons in the system under test (SUT)
4. The vendor is using the same version of SUT during self-test and live test
5. Approval requires that the SUT is approved for sending a FHIR Acknowledgement (DK: Kvittering).

## Documentation of self-test

**Self-test**

**Prior to the test, the vendor must have performed self-test, which is approved by MedCom.**

The self-test is documented by the vendor completing this test protocol.

For self-tests, only the following column must be completed by the vendor:

* [Test data]: is filled in with the file name(s) which are uploaded and downloaded
* [Actual result]: is filled in with the results of the self-test and relevant descriptions.

Other columns are reserved for MedCom.

**During the self-test the vendor must document the test results by saving relevant files and screen dumps, and subsequently send these in a combined ZIP file (together with the completed test protocol) to** [**fhir@medcom.dk**](mailto:fhir@medcom.dk)**.**

All files and screen dumps must be named with:

* Standard name
* The number of the relevant test step
* Consecutive letter
* Whether SUT is sender (S) or recipient (R) of the standard
* File type

*Example: CareCommunication \_3.4\_A\_R.xml*

## Background material

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version** | **Link/reference** | **Description** |
| CareCommunication – documentation site |  | <https://medcomdk.github.io/dk-medcom-carecommunication/> | Documentation site with references to all relevant documentation, including:   * Clinical guidelines for application (Sundhedsfaglige retningslinjer for anvendelse) * Use cases * Technical specifications |
| Implementation Guide |  | <https://medcomfhir.dk/ig/carecommunication/> | The FHIR technical guidelines for the standard. |
| Governance for MedCom FHIR |  | <https://medcomdk.github.io/MedCom-FHIR-Communication/> | Governance for MedCom’s FHIR standards, which describes general rules for all MedCom standards and specific rules for this standard, as well as for sending. |
| SOP for MedCom’s test and certification |  | <http://svn.medcom.dk/svn/qms/Offentlig/SOPer/SOP-7.2-MedComs%20test%20og%20certificering_godkendelse.docx> | Description of test and certification of MedCom standards and other tests courses. |
| Conversion between formats |  | <https://medcomdk.github.io/dk-medcom-carecommunication/#3-conversion-service> | Is under clarification but is planned in the context of the VANS cooperation. The plan is to ensure conversion from FHIR to OIOXML and OIOXML to FHIR. Attached files and Acknowledgements are also handled via the conversion service. |

## Test examples and test persons

|  |  |  |
| --- | --- | --- |
| **Name** | **Link/reference** | **Description** |
| Test examples | <http://medcomfhir.dk/ig/carecommunicationtestscripts/testexamples.html> | Test examples used during the test and certification. |
| Overview of test persons | <https://www.medcom.dk/opslag/koder-tabeller-ydere/tabeller/nationale-test-cpr-numre> | Overview of national test CPR numbers which can be used during the test.  **Please notice**: During the test, the vendor must be able to use any of the test persons on the list. |

## Test tools

|  |  |  |
| --- | --- | --- |
| **Name** | **Link/reference** | **Description** |
| FHIR server with MedCom profiles | [insertLink will be provided. | Public server that validates against MedCom's FHIR profiles. It is permitted to use the server for testing the upload/download of FHIR resources. |
| TouchStone | <https://touchstone.aegis.net/touchstone/> | Test tool for testing the FHIR standard.  The vendor can get access to TouchStone as an organisation - either through a license that MedCom supplies (inquiry at [fhir@medcom.dk](mailto:fhir@medcom.dk)), or a license that the vendor has acquired itself.  Find [instructions for TouchStone](https://medcomdk.github.io/MedComLandingPage/assets/documents/TouchStoneGettingStarted.html) here. |
| Touchstone test scripts | <https://touchstone.aegis.net/touchstone/conformance/current?suite=FHIR4-0-1-CareCommunication-v300-Receive-Client>  and  <https://medcomfhir.dk/ig/carecommunicationtestscript/> | Test scripts relevant for the standard.  These are not mandatory to execute.  Find [instructions to TouchStone here](https://medcomdk.github.io/MedComLandingPage/assets/documents/TouchStoneGettingStarted.html). |

## Test result

The result for each test step is categorised based on the table below:

| **Marking** | **F1** | **F2** | **F3** | **F4** | **OK** | **Not relevant** |
| --- | --- | --- | --- | --- | --- | --- |
| **Evaluation** | **Critical** | **Serious** | **Significant** | **Less significant** | **Approved** | **Not an error** |

To get the test and certification approved, the test protocol must consist exclusively of [F4] as well as [OK] results. Thus, all [F1], [F2] and [F3] must, be fixed prior to final approval.

When a test step isn’t relevant for the test course, it is noted with ‘Not relevant’.

Approval requires that SUT is approved for sending a FHIR acknowledgement (DK: Kvittering)

For further information, please read: [MedCom’s test and](#TestCertificering) certification.

# Information about vendor, system under test (SUT) and test result information

## Information about the vendor

This table must be completed by **the vendor** prior to the test.

|  |  |
| --- | --- |
| Company | Completed by vendor |
| Address | Completed by vendor |
| Contact person | Completed by vendor |
| Telephone | Completed by vendor |
| E-mail | Completed by vendor |

## Information about the system under test (SUT)

This table must be completed by **the vendor** prior to the test.

|  |  |
| --- | --- |
| System | Completed by vendor |
| Version | Completed by vendor |
| Description | Completed by vendor |
| Test type | Self-test  Final test/certification |

## Information about test results

This table must be completed by MedCom when the test has been completed.

|  |  |
| --- | --- |
| Test date | 2023-01-01 |
| Test location | Completed by MedCom |
| Approved | Yes  No |
| Remarks | Completed by MedCom |
| Carried out by | Completed by MedCom. The name of the MedCom responsible (initials) for this test is inserted |

# The test

This section describes the requirements which SUT must meet before final approval.

The test is divided into three sections:

1. Test of TouchStone test scripts
2. Test of requirements for content and flow/workflows
3. Test of general technical requirements

Test participants will be asked to complete tests as described in the tables.

## Documentation of the test

**Documentation of the test**

As valid documentation, the test participant or the test manager must document completion by continuous screen dumps (.png/.jpeg) and/or files/log files (.xml/.json). **Before the test, it is agreed on who is responsible for this.**

The following applies:

* The files must be viewable in a standard tool and must not require further processing by MedCom
* All files and screen dumps must be named with:
  + The name of the standard
  + The number of the relevant test setup
  + Consecutive letter
  + Whether SUT is the sender (S) or recipient (R) of the standard
  + File type

*Example:* *CareCommunication\_3.4\_A\_S.xml, CareCommunication\_ 3.4\_B \_S.xml*

If the vendor has documented the test themselves, the files must be sent in a ZIP file to [fhir@medcom.dk](mailto:fhir@medcom.dk).

## Test of TouchStone test scripts

The purpose of these tests is to ensure that SUT generates the message technically correct and complies with the rules in the [Implementation Guide](#_Baggrundsmaterialer). This test step is optional to go through.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | **Optional test step:**  Run all test scripts for use cases and user flows in TouchStone. |  | All test scripts completed without errors. |  | Choose |

## Test of requirements to content and flow/workflows

The purpose of these tests is to ensure that the standard is implemented with satisfactory quality, i.e. that implementation meets the business requirements for flow and content as described in the clinical guidelines and the [use case material](#_Baggrundsmaterialer). These test steps are mainly for the user interface.

The table below lists the use cases which are tested in relation to content and flow/workflows. The table also shows the direct references to the use cases in the [use case material](#_Baggrundsmaterialer).

|  |  |  |
| --- | --- | --- |
| [**Use case**](#_Baggrundsmaterialer) | **Description** | **Section** |
| R1 | Receive a CareCommunication | 3.3.1 |
| R2 | Receive a reply to a CareCommunication | 3.3.2 |
| R3 | Receive a forwarded CareCommunication | 3.3.3 |
| R.CANC | Receive a cancellation to a CareCommunication received previously | 3.3.4 |
| R.CORR | Receive a correction to a CareCommunication received previously | 3.3.5 |

Table 1: Table listing the use cases which must be tested

### R1: Receive a CareCommunication

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | Load a CareCommunication test example. | CareCommunication\_Ex\_ new-A | CareCommunication test example is loaded. |  | Choose |
|  | Describe how the SUT notifies the user that a CareCommunication has been received. |  | User is notified that a new CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT clearly indicates that the message is a new message.  *It is recommended that the SUT clearly indicates whether the message is new, a reply or a forwarded message.* |  | It is clearly indicated in the user interface that the message is a new message. |  | Choose |
|  | Demonstrate that the user opens the received CareCommunication. |  | The user has opened the received CareCommunication. |  | Choose |
|  | Demonstrate that the SUT shows relevant information for the user |  | The user can see the following information from the message.   * patient id and name, * category, * message segment including message text and signature, * sender, * receiver.   If included:   * specific sender * specific recipient * topic * priority * attachments |  | Choose |
|  | Demonstrate that the SUT shows message segments with attached files, including titles of the files and (if filled in) name of author and time of creation of the attached files. |  | The user can read all content of the message, including attached files. |  | Choose |
|  | Demonstrate that the SUT can load and show all allowed file types attached to the message. |  | The user can open and read all attached file types which are allowed. The list of [allowed file types can be accessed via the IG](https://medcomfhir.dk/ig/terminology/ValueSet-medcom-core-attachmentMimeTypes.html). |  | Choose |

### R2: Receive a reply to a CareCommunication

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | Load the test example for receiving a reply to a CareCommunication | CareCommunication\_Ex\_ reply-A | CareCommunication test example is loaded. |  | Choose |
|  | Describe how the SUT notifies the user that a CareCommunication has been received. |  | User is notified that a new CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT clearly indicates to the user that a reply to a message sent previously has been received.  *It is recommended that the SUT clearly indicates to the user that the message is a new reply.* |  | It is clearly indicated in the user interface that a new message has been received. |  | Choose |
|  | Demonstrate that the user opens the received message. |  | The user has opened the received message. |  | Choose |
|  | Demonstrate that the SUT shows relevant information for the user. |  | The user can see the following information from the message.   * patient id and name, * category, * new message segment including message text and signature, * previously send message segment including message text and signature, * sender, * receiver.   If included:   * specific sender * specific recipient * topic * priority * attachments |  | Choose |
|  | Demonstrate that the SUT shows message segments with attached files, including titles of the files and (if filled in) name of author and time of creation of the attached files. |  | The user can read all content of the message, including attached files. |  | Choose |
|  | Demonstrate that the SUT can load and show all allowed file types attached to the message. |  | The user can open and read all attached file types which are allowed. The list of [allowed file types can be accessed via the IG](https://medcomfhir.dk/ig/terminology/ValueSet-medcom-core-attachmentMimeTypes.html). |  | Choose |

### R3: Receive a forwarded CareCommunication

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | Load the test example for receiving a forwarded CareCommunication. | CareCommunication\_Ex\_ forward-A | CareCommunication test example is loaded. |  | Choose |
|  | Describe how the SUT notifies the user that a CareCommunication has been received. |  | User is notified that a new CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT clearly indicates to the user that the message is a forwarded message.  *It is recommended that the SUT clearly indicates to the user that the message is a forwarded message.* |  | It is clearly indicated in the user interface that a forwarded message has been received. |  | Choose |
|  | Demonstrate that the user opens the received CareCommunication. |  | The user has opened the received CareCommunication. |  | Choose |
|  | Demonstrate that the SUT shows relevant information for the user. |  | The user can see the following information from the message.   * patient id and name, * category, * new message segment including message text and signature, * previously send message segment including message text and signature, * sender, * receiver.   If included:   * specific sender * specific recipient * topic * priority * attachments |  | Choose |
|  | Demonstrate that the SUT shows message segments with attached files, including titles of the files and (if filled in) name of author and time of creation of the attached files. |  | The user can read all content of the message, including attached files. |  | Choose |
|  | Demonstrate that the SUT can load and show all allowed file types attached to the message. |  | The user can open and read all attached file types which are allowed. The list of [allowed file types can be accessed via the IG](https://medcomfhir.dk/ig/terminology/ValueSet-medcom-core-attachmentMimeTypes.html). |  | Choose |

### R.CANC: Receive an cancellation to a CareCommunication received previously

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | Load test example for cancellation. | CareCommunication\_Ex\_ cancel-A | Test example for cancellation is loaded. |  | Choose |
|  | Demonstrate that the SUT loads the cancellation correctly and link it to a CareCommunication already received. |  | A cancellation is correctly loaded and linked to a CareCommunication already received. |  | Choose |
|  | Demonstrate that the SUT notifies the user that a cancellation has been received and that it is clear which CareCommunication the cancellation concerns. |  | The user is notified that a specific CareCommunication received earlier has been cancelled. |  | Choose |
|  | Demonstrate that the SUT loads and shows a short text explaining why the received message was cancelled. |  | It is clear to the user what the cancellation concerns and why the message was cancelled by the sender. |  | Choose |
|  | Demonstrate that the status of the entire message thread is ‘cancelled’. |  | The entire message thread has the status ‘cancelled’. |  | Choose |
|  | Demonstrate the user no longer can reply, forward or correct any of the messages in the message thread. |  | The user cannot reply, forward or correct any of the messages in the message thread. |  | Choose |
|  | **Cancellation of correction**  Load the test data file which is a cancellation of a correction.  Demonstrate that the user can clearly see that the received correction has been cancelled. | CareCommunication\_Ex\_ cancel-B | A cancellation is correctly loaded and linked to a correction received. The status of the entire message thread is ‘cancelled’ |  | Choose |
|  | **Cancellation of a message after a correction has been received.**  Load the test data file which is a cancellation of a CareCommunication which has been corrected.  Demonstrate that the user can clearly see that the original message has been cancelled and that the received correction therefore no longer is valid and thus cancelled. | CareCommunication\_Ex\_ cancel-C | A cancellation is correctly loaded and linked to a received CareCommunication, and the correction linked to the message. |  | Choose |

### R.CORR: Receive a correction to a CareCommunication received previously

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
|  | Load test example for correction. | CareCommunication\_Ex\_ correction-A | Test example for correction is loaded. |  | Choose |
|  | Demonstrate that the SUT loads the correction correctly and links the message to a CareCommunication already received. |  | A correction is loaded correctly and linked to CareCommunication already received. |  | Choose |
|  | Demonstrate that the SUT notifies the user of a received correction with a clear indication of which CareCommunication the correction concerns. |  | The user is notified that a specific CareCommunication received earlier is corrected. |  | Choose |
|  | Demonstrate in the message text that the user can see the message being corrected and the actual correction. |  | The user can see the message being corrected and the actual correction. |  | Choose |
|  | Demonstrate that the status of the entire message thread is ‘corrected’ and that the user no longer can reply or forward any of the messages in the thread. |  | The status of the entire message thread is ‘corrected’ and the user cannot reply or forward the messages in the thread |  | Choose |

## Test of general technical requirements

The purpose of these test steps is to ensure that the technical sending of a CareCommunication is implemented with satisfactory quality, i.e., that it meets the governance for message communication on a general level as well as governance for CareCommunication as described in 1.4.

### Formatted text

Complete the test step in the following which best represents SUT functionality

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | *Complete this test step if the SUT* ***supports*** *formatting of free text in the message’s text box in accordance with* [*MedCom’s subset of XHTML for formatting*](https://medcomdk.github.io/dk-medcom-core/assets/documents/MedComCore-Styling_the_XHTML.html)*.*  Load the test data and demonstrate that the SUT can show formatted message text and table(s) | CareCommunication\_Ex\_ Tek-new-formatted | The user can see all the message text and the associated formatting. |  | Choose |
|  | *Complete this test step if the SUT* ***does not*** *support formatting of free text in the message’s text box in accordance with* [*MedCom’s subset of XHTML for formatting*](https://medcomdk.github.io/dk-medcom-core/assets/documents/MedComCore-Styling_the_XHTML.html)*.*  Load the test data and demonstrate that the SUT can remove the formatting and display the message text without the possibility of misinterpretation | CareCommunication\_Ex\_ Tek-new-formatted | The user can see the message text without formatting. |  | Choose |
|  | *Complete this test step if the SUT* ***supports a part of*** *formatting of free text in the message’s text box in accordance with* [*MedCom’s subset of XHTML for formatting*](https://medcomdk.github.io/dk-medcom-core/assets/documents/MedComCore-Styling_the_XHTML.html)*.*  Explain which part of the formatting SUT supports. |  | Description of which parts of formatting SUT supports. |  | Choose |
|  | Load the test data and demonstrate that the SUT can show formatted message text and remove the formatting where relevant. | CareCommunication\_Ex\_ Tek-new-formatted | The user can see all the message text and the associated formatting for the parts SUT supports and the remaining text without formatting. |  | Choose |

### Priority

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | Load the test data and demonstrate that SUT displays priority, when it is included in the message. | CareCommunication\_Ex\_ Tek-new-priority | Priority is displayed for the user. |  | Choose |

### Display organization name for sender and receiver

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | *In FHIR messages it is not required to include the Organization.name for the sender and receiver. Therefore, SUT must use SOR to get this information.*  Load the test data without Organization.name on sender and receiver and show that SUT get the names from SOR. | CareCommunication\_Ex\_ Tek-new-noOrgName | SUT uses SOR to get the sender and receiver name based on the SOR-id. SUT displays correct name for the sender and receiver. |  | Choose |

### Support of XML and JSON

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | *FHIR messages may be send in XML or JSON. SUT must be able to handle both.*  Load the test data in XML-format and show that SUT can display the content. | CareCommunication\_Ex\_ Tek-new-xml | SUT displays the CareCommunication received in XML format. |  | Choose |
|  | Load the test data in JSON-format and show that SUT can display the content. | CareCommunication\_Ex\_ Tek-new-json | SUT displays the CareCommunication received in JSON format. |  | Choose |

### Cancellations and corrections

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | **Cancellation**  In case a cancellation to a message, which has already been forwarded, is received, the forwarded message should not automatically be cancelled.  Demonstrate or explain how the SUT handles the above. |  | When a cancellation to a message, which has already been forwarded, is received, the forwarded message is not automatically cancelled. |  | Choose |
|  | **Correction**  In case a correction to a message, which has already been forwarded, is received, the forwarded message should not automatically be corrected.  Demonstrate or explain how the SUT handles the above. |  | When a correction to a message, which has already been forwarded, is received, the forwarded message is not automatically corrected. |  | Choose |

### Parallel sent CareCommunications

The following test step concerns correct handling of parallel sent replies. Parallel means that two corresponding parties replies to the same CareCommunication more or less synchronous and/or due to delays in the systems or on the VANS network the CareCommunications. .

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | ***Parallel sent replies:***  Load test data and sent a reply. SUT must create two communication-threads, one with each reply but both including the previously sent CareCommunication. | CareCommunication\_Ex\_ Tek-reply-parallel-A | SUT creates two communication threads for each of the replies. |  | Choose |
|  | Demonstrate that SUT makes it visible to the user that a CareCommunication is new. |  | It is visible to the user that there is an unread CareCommunication. |  | Choose |
|  | ***Received reply parallel with sent correction/cancellation:***  Load test data and sent a correction/cancellation. SUT includes the reply in the cancelled/corrected CareCommunication thread or a separate CareCommunication thread. | CareCommunication\_Ex\_ Tek-reply-parallel-B | SUT displays the reply in the same CareCommunication thread as the cancelled/corrected or in a separate thread. |  | Choose |
|  | Explain if SUT displays the cancelled/corrected CareCommunication in a separate thread or not. |  | Description of how SUT displays the cancelled/corrected CareCommunication. |  |  |
|  | Demonstrate that SUT makes it visible to the user that a CareCommunication is new. |  | It is visible to the user that there is an unread CareCommunication. |  | Choose |
|  | Demonstrate that the CareCommunication thread(s) appears as corrected or cancelled. |  | The CareCommunication thread(s) appears as corrected or cancelled. |  | Choose |
|  | Demonstrate that the user no longer can communicate in the thread(s). |  | The user can no longer communicate in the thread(s). |  | Choose |
|  | ***Received correction/cancellation parallel with sent reply:***  Load test data and sent a reply. SUT includes the cancelled/corrected in thread with the reply or a separate CareCommunication thread. | CareCommunication\_Ex\_ Tek-correction-parallel-A or  CareCommunication\_Ex\_ Tek-cancel-parallel-A | SUT display the cancelled/corrected CareCommunication in the same thread as the reply or in a separate thread. |  | Choose |
|  | Demonstrate that SUT makes it visible to the user that a correction/cancellation is received. |  | It is visible to the user that there is an unread CareCommunication. |  | Choose |
|  | Demonstrate that the CareCommunication thread(s) appears as corrected or cancelled. |  | The entire CareCommunication thread(s) appears as corrected or cancelled. |  | Choose |
|  | Demonstrate that the user no longer can communicate in the thread(s). |  | The user can no longer communicate in the thread(s). |  | Choose |
|  | **Parallel sent corrections/cancellations:**  Load test data and sent a correction/cancellation. SUT includes the correction/cancellation in the cancelled/corrected CareCommunication thread or in two separate threads. | CareCommunication\_Ex\_ Tek-correction-parallel-B or  CareCommunication\_Ex\_ Tek-cancel-parallel-B | SUT display the two cancelled/corrected CareCommunication in the same thread or in two separate threads. |  | Choose |
|  | Demonstrate that SUT makes it visible to the user that a correction/cancellation is received. |  | It is visible to the user that there is an unread CareCommunication. |  | Choose |
|  | Demonstrate that the CareCommunication thread(s) appear as corrected or cancelled.  A cancellation overrules a correction. |  | The CareCommunication thread(s) appears as corrected or cancelled. |  | Choose |
|  | Demonstrate that the user no longer can communicate in the thread(s). |  | The user can no longer communicate in the thread(s). |  | Choose |

### Receive CareCommunication and send FHIR Acknowledgement

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | Describe how FHIR messages are loaded into the SUT. For example, how is data loaded into the SUT – via mapping to internal format or to own FHIR infrastructure? |  | For example: FHIR messages are used directly or processed. |  | Choose |
|  | **Acknowledgement**  Load test data and demonstrate that the SUT sends a FHIR Acknowledgement in return (DK: Kvittering). | CareCommunication\_Ex\_ Tek-new-ack | SUT returns a FHIR Acknowledgement (DK: Kvittering) |  | Choose |

### Loading of FHIR messages where the received messages are not in the same order as they were sent

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | **A cancellation is received before the new message.**  Load test data, which is an cancellation of a new CareCommunication and demonstrate that the message is cancelled in the SUT. | CareCommunication\_Ex\_ Tek-cancel-order-A | The cancelled message appears as cancelled in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |
|  | Load test data for a new CareCommunication and demonstrate that the message appears as cancelled in the SUT. | CareCommunication\_Ex\_ Tek-new-order-A | The cancelled message appears as annulled in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |
|  | **A correction is received before the new message.**  Load the test data, which is a correction to a new CareCommunication, and demonstrate that the message appears as corrected in the SUT. | CareCommunication\_Ex\_ Tek-correction-order-B | The corrected message appears as corrected in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |
|  | Load test data for a new CareCommunication and demonstrate that the message appears as corrected in the SUT. | CareCommunication\_Ex\_ Tek-new-order-B | The corrected message appears as corrected in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |
|  | **A reply is received before the new message.**  Load the test data, which is a reply to a new CareCommunication, and demonstrate that the message appears as replied in the SUT. | CareCommunication\_Ex\_ Tek-reply-order-C | The replied message appears as replied in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |
|  | Load test data for a new CareCommunication and demonstrate that the message appears as replied in the SUT. | CareCommunication\_Ex\_ Tek-new-order-C | The replied message appears as replied in the user interface.  The SUT returns a FHIR Acknowledgement (DK: Kvittering) to the correct receiver. |  | Choose |

### Handling of doublets

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | **Positive doublet – identical Bundle.id:**  Load test data and demonstrate that the CareCommunication is loaded and accessible to the SUT user. | CareCommunication\_Ex\_ Tek-new-duplicate-A | The SUT user can see that a CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a positive FHIR Acknowledgement (Kvittering). |  | The SUT has acknowledged positively for the CareCommunication message and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |
|  | Load test data again and demonstrate that the CareCommunication, which is a duplet, is ignored and that the SUT user still only sees one CareCommunication message. | CareCommunication\_Ex\_ Tek-new-duplicate-A | The SUT user still only sees that one CareCommunication message has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a positive FHIR Acknowledgement (Kvittering) for the duplet. |  | The SUT has acknowledged positively for the doublet and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |
|  | **Positive doublet – different Bundle.id:**  Load test data and demonstrate that the CareCommunication is loaded and accessible to the SUT user. | CareCommunication\_Ex\_ Tek-new-duplicate-B1 | The SUT user can see that a CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a positive FHIR Acknowledgement (Kvittering). |  | The SUT has acknowledged positively for the CareCommunication message and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |
|  | Load test data again and demonstrate that the CareCommunication, which is a duplet, is ignored and that the SUT user still only sees one CareCommunication message. | CareCommunication\_Ex\_ Tek-new-duplicate-B2 | The SUT user still only sees that one CareCommunication message has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a positive FHIR Acknowledgement (Kvittering) for the duplet. |  | The SUT has acknowledged positively for the doublet and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |
|  | **Negative doublet:**  Load test data and demonstrate that the CareCommunication is loaded and accessible to the SUT user. | CareCommunication\_Ex\_ Tek-new-duplicate-C | The SUT user can see that a CareCommunication has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a negative FHIR Acknowledgement (Kvittering). |  | The SUT has acknowledged negatively for the CareCommunication message and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |
|  | Load test data again and demonstrate that the CareCommunication, which is a doublet, is ignored and that the SUT user still only sees one CareCommunication message. | CareCommunication\_Ex\_ Tek-new-duplicate-C | The SUT user still only sees that one CareCommunication message has been received. |  | Choose |
|  | Demonstrate that the SUT has sent a negative FHIR Acknowledgement (Kvittering) for the duplet. |  | The SUT has acknowledged negatively for the doublet and sent a FHIR Acknowledgement (Kvittering) to the correct receiver. |  | Choose |

### Handling of invalid messages

| **Test step #** | **Action** | **Test data/test person** | **Expected result** | **Actual result** | **MedCom assessment** |
| --- | --- | --- | --- | --- | --- |
|  | Load test data and demonstrate how the SUT handles invalid messages, e.g. missing valid “Communcation.category” | CareCommunication\_Ex\_ Tek-new-invalid | The SUT returns a FHIR Acknowledgement (Kvittering) where the error is described. |  | Choose |
|  | SUT does not display the invalid CareCommunication. |  | The SUT does not show the invalid message. |  | Choose |