# ETSI GS CDM 007-2 V1.1.1 (2024-07)


# Common information sharing environment service and Data Model (CDM); Testing; Conformance test specifications for CISE; Part 2: Test Suite Structure and Test Purposes (TSS & TP); Release 1

***Disclaimer***

The present document has been produced and approved by the european Common information sharing environment service and Data Model (CDM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG. It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/CDM-0013

Keywords
data sharing, testing, TSS&TP

***ETSI***

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

# Contents

# Intellectual Property Rights

## Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

## Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

# Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) european Common information sharing environment service and Data Model (CDM).

The present document is part 2 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.6].

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# 1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS & TP) for the CDM service APIs for the exchange of messages complying with the CDM Data Model, in accordance with the relevant guidance given in ISO/IEC 9646-7 [i.4].

The ISO standards for the methodology of conformance testing (ISO/IEC 9646-1 [i.1] and ISO/IEC 9646-2 [i.2]) as well as the ETSI rules for conformance testing (ETSI ETS 300 406 [i.5]) are used as a basis for the test methodology.

# 2 References

## 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS CDM 004: "Common information sharing environment service and Data Model (CDM); Service Model; Release 1".

[2] ETSI GS CDM 005: "Common information sharing environment service and Data Model (CDM); Data Model; Release 1".

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ISO/IEC 9646-1 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 1: General concepts".

[i.2] ISO/IEC 9646-2 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 2: Abstract Test Suite specification".

[i.3] ISO/IEC 9646-6 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 6: Protocol profile test specification".

[i.4] ISO/IEC 9646-7 (1995): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 7: Implementation Conformance Statements".

[i.5] ETSI ETS 300 406 (1995): "Methods for testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[i.6] ETSI GS CDM 007-1: "Common information sharing environment service and Data Model (CDM); Testing; Conformance test specifications for CISE; Part 1: Test requirements and Protocol Implementation Conformance Statement (PICS) proforma; Release 1".

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in ETSI GS CDM 004 [1], ETSI GS CDM 005 [2], ISO/IEC 9646-6 [i.3] and ISO/IEC 9646-7 [i.4] apply.

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

| | |
|---|---|
| BV | Valid Behaviour tests |
| CDM | CISE Data Model |
| CISE | Common Information Sharing Environment |
| HTTP | Hypertext Transfer Protocol |
| IUT | Implementation Under Test |
| PICS | Protocol Implementation Conformance Statement |
| TP | Test Purpose |
| TSS | Test Suite Structure |

# 4 Test Suite Structure (TSS)

## 4.1 Structure for CDM API tests

Table 1 shows the CDM API Test Suite Structure (TSS) defined for conformance testing.

**Table 1: TSS for CDM**

| Root | Group | SubGroup | Category |
|---|---|---|---|
| CDM | NM EA Interface | PullRequest | Valid and Invalid |
| | | PullRequest Multicast | Valid |
| | | PullRequest Discovery | Valid and Invalid |
| | | Subscribe | Valid and Invalid |
| | | Subscribe Lookup | Valid and Invalid |
| | | Unsubscribe | Valid and Invalid |
| | | Asynchronous Acknowledgement | Valid |
| | | Feedback | Valid |
| | NM EA Interface | PullRequest | Valid and Invalid |
| | | PullRequest Multicast | Valid |
| | | PullRequest Discovery | Valid and Invalid |
| | | Subscribe | Valid and Invalid |
| | | Subscribe Lookup | Valid and Invalid |
| | | Unsubscribe | Valid and Invalid |
| | | Asynchronous Acknowledgement | Valid |
| | | Feedback | Valid |

| Root | Group | SubGroup | Category |
|---|---|---|---|
| | Adaptor | PullRequest | Valid and Invalid |
| | | PullRequest Multicast | Valid |
| | | PullRequest Discoveryr | Valid and Invalid |
| | | Subscribe | Valid and Invalid |
| | | Subscribe Lookup | Valid and Invalid |
| | | Unsubscribe | Valid and Invalid |
| | | Asynchronous Acknowledgement | Valid |
| | | Feedback | Valid |
| | Security | N/A | Valid and Invalid |

# 5 CDM Security

## 5.1 Overview

### 5.1.1 Certificates content

#### 5.1.1.1 Certificate in the past

In addition of regular signing certificate, one expired certificate is used to verify that the IUT discard message signed with this invalid certificate.

#### 5.1.1.2 Certificate in the future

In addition of regular signing certificate, one certificate valid in the future is used to verify that the IUT discard message signed with this invalid certificate.

# 6 Test Purposes (TP)

## 6.1 Introduction

### 6.1.1 TP definition conventions

The TP definition is built according to ISO/IEC 9646-6 [i.3].

### 6.1.2 TP Identifier naming conventions

The identifier of the TP is built according to Table 2.

**Table 2: TP naming convention**

| Identifier | TP_<root>_<tgt>_<gr>_<sub-gr> <sn>_<x> | Sub-Group | Category |
|---|---|---|---|
| TP | <root> = root | CDM | CISE Conformance Test |
| | <tgt> = target | NODE | CISE Node/Adaptor interface |
| | | NODE_I2 | CISE Node/Gateway interface |
| | | ADAPTOR | Adaptor |
| | | SECURITY | Security |
| | <gr> = group | PULL_REQUEST | PullRequest communication |
| | | PULL_MULTI_PULL_REQ | PullRequest multicast communication |
| | | PULL_DISC | PullRequest Discovery |
| | | SUB | Subscription |
| | | SUB_LOOK | Subscribers lookup |
| | | SUB_DELETE | Subscription deletion |
| | | PUSH_NOTIFY | Notification communication |
| | | ACK | Asynchronous acknowledgement |
| | | FEEDBACK | Feedback communication |
| | <sn> = test purpose sequential number | | 01 to 99 |
| | <x> = category | BV | Valid Behaviour tests |
| | | BO | Invalid Behaviour Tests |

### 6.1.3 Rules for the behaviour description

The description of the TP is built according to ISO/IEC 9646-6 [i.3].

Being in the "Initial State" refers to the starting point of the initial device configuration. There are no pending actions, no instantiated buffers or variables, which could disturb the execution of a test.

### 6.1.4 Sources of TP definitions

All TPs have been specified according to ETSI GS CDM 004 [1] and ETSI GS CDM 005 [2].

### 6.1.5 Mnemonics for PICS reference

To avoid an update of all TPs when the PICS document is changed, Table 3 introduces mnemonics name and the correspondence with the real PICS item number. The 'PICS item' shall be used to determine the test applicability.

**Table 3: Mnemonics for PICS reference**

| N. | Mnemonic | PICS item |
|---|---|---|
| 1 | IUT_NODE | ETSI GS CDM 007-1 [i.6] A.1/1 |
| 2 | IUT_ADAPTOR | ETSI GS CDM 007-1 [i.6] A.1/2 |
| 3 | CDM_PULL | ETSI GS CDM 007-1 [i.6] A.2/1.1 |
| 4 | CDM_PUSH | ETSI GS CDM 007-1 [i.6] A.2/1.2 |
| 5 | CDM_SUBSCRIBE | ETSI GS CDM 007-1 [i.6] A.2/1.3 |
| 6 | CDM_FEEDBACK | ETSI GS CDM 007-1 [i.6] A.2/1.4 |
| 7 | CDM_VESSEL_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.11 |
| 8 | CDM_ACTION_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.1 |
| 9 | CDM_ANOMALY_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.2 |
| 10 | CDM_CERTIFICATE_DOCUMENT_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.3 |
| 11 | CDM_INCIDENT_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.4 |
| 12 | CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.5 |
| 13 | CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.6 |
| 14 | CDM_METEO_OCEANOGRAPHIC_CONDITION_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.7 |
| 15 | CDM_ORGANIZATION_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.8 |

| N. | Mnemonic | PICS item |
|---|---|---|
| 16 | CDM_RISK_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.9 |
| 17 | CDM_CARGO_SERVICE | ETSI GS CDM 007-1 [i.6] A.3/2.10 |

# 6.2 EI NA Interface

## 6.2.1 Pull Request

### 6.2.1.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_01_01" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest for Vessel service - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
```

```
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_01_02" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with timeout on response - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
```

```
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
                }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_01_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK PullResponse response with a ResultCode 'InvalidRequestObject' when receiving a PullRequest for a unknown vessel |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel indicating value empty,
          PullType indicating value Request,
          DiscoveryProfiles containing
            country indicating value COUNTRY,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
```

```
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value InvalidRequestObject,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_01_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with filter - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered several vessel containing
  VESSEL_1 containing
    IMONumber indicating value VESSEL_IMO_NUMBER_1,
    Name indicating value VESSEL_NAME_1,
    Location indicating value VESSEL_LOCATION_1,
    ShipType indicating value VESSEL_SHIP_TYPE_1,
    NetTonnage indicating value VESSEL_NET_TONNAGE_GREATHER_THAN,
  VESSEL_2 containing
    IMONumber indicating value VESSEL_IMO_NUMBER_2,
    Name indicating value VESSEL_NAME_2,
    Location indicating value VESSEL_LOCATION_2,
    ShipType indicating value VESSEL_SHIP_TYPE_2,
    NetTonnage indicating value VESSEL_NET_TONNAGE_LESS_THAN
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value vesselService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value vesselService,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER_1,
            Name indicating value VESSEL_NAME_1,
            Location indicating value VESSEL_LOCATION_1,
            ShipType indicating value VESSEL_SHIP_TYPE_1,
            NetTonnage indicating value VESSEL_NET_TONNAGE_1,
        PayloadSelector containing
          Selectors containing // Vessels with tonnage > VESSEL_NET_TONNAGE
            Selector indicating value "//Vessel[1]/NetTonnage",
            Operator indicating value GREATER_THAN,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
    status_code set to "200 OK",
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value vesselService,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value vesselService,
          ServiceOperation indicating value Pull,
        Payload containing
          vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER_1,
            Name indicating value VESSEL_NAME_1,
            Location indicating value VESSEL_LOCATION_1,
            ShipType indicating value VESSEL_SHIP_TYPE_1,
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
              }
```

| TP Id | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_01" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving an inconsistent PullRequest |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Push, // Push instead of pull
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Push, // Push instead of pull
          not Payload,
          PullType indicating value Request,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value InvalidRequestObject,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

| TP Id | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_02" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with an inconsistent Service Operation type |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
```

```
  ShipType indicating value VESSEL_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Push, // Inconsistent Service Operation type
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER_1,
              Name indicating value VESSEL_NAME_1,
              Location indicating value VESSEL_LOCATION_1,
              ShipType indicating value VESSEL_SHIP_TYPE_1,
              NetTonnage indicating value VESSEL_NET_TONNAGE_1,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value InvalidRequestObject,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_03" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with an unknown serviceID |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID_UNKNOWN, // Unknown service ID
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER_1,
              Name indicating value VESSEL_NAME_1,
              Location indicating value VESSEL_LOCATION_1,
              ShipType indicating value VESSEL_SHIP_TYPE_1,
              NetTonnage indicating value VESSEL_NET_TONNAGE_1,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value serviceTypeNotSupported,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with a date/time far in past (one month late) |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME - 30 days, // Date/time far in past one
month late
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
```

```
          ServiceOperation indicating value Pull,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER_1,
            Name indicating value VESSEL_NAME_1,
            Location indicating value VESSEL_LOCATION_1,
            ShipType indicating value VESSEL_SHIP_TYPE_1,
            NetTonnage indicating value VESSEL_NET_TONNAGE_1,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value timestampError,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_05" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with a date/time far in future (one month) |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME - 30 days, // Date/time far in future one
month
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER_1,
              Name indicating value VESSEL_NAME_1,
              Location indicating value VESSEL_LOCATION_1,
              ShipType indicating value VESSEL_SHIP_TYPE_1,
              NetTonnage indicating value VESSEL_NET_TONNAGE_1,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
```

```
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value timestampError,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_01_06" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with an invalid filter (Selector entry does not exist) |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered several vessel containing
  VESSEL_1 containing
    IMONumber indicating value VESSEL_IMO_NUMBER_1,
    Name indicating value VESSEL_NAME_1,
    Location indicating value VESSEL_LOCATION_1,
    ShipType indicating value VESSEL_SHIP_TYPE_1,
    NetTonnage indicating value VESSEL_NET_TONNAGE_GREATHER_THAN,
  VESSEL_2 containing
    IMONumber indicating value VESSEL_IMO_NUMBER_2,
    Name indicating value VESSEL_NAME_2,
    Location indicating value VESSEL_LOCATION_2,
    ShipType indicating value VESSEL_SHIP_TYPE_2,
    NetTonnage indicating value VESSEL_NET_TONNAGE_LESS_THAN
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER_1,
              Name indicating value VESSEL_NAME_1,
              Location indicating value VESSEL_LOCATION_1,
              ShipType indicating value VESSEL_SHIP_TYPE_1,
              NetTonnage indicating value VESSEL_NET_TONNAGE_1,
          PayloadSelector containing
            Selectors containing // Vessels with tonnage > VESSEL_NET_TONNAGE
              Selector indicating value "//Vessel[2]/NetTonnage", // Selector does not exist
              Operator indicating value GREATER_THAN,
          PullType indicating value Request,
```

```
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value badRequest,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

### 6.2.1.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_02_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                 }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value nonSpecified,
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
```

```
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_02_02" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with timeout on response |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value "Any", // FIXME
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
```

```
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_02_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with inconsistent Location |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value Pull,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value "Any", // FIXME
              Priority indicating value High,
              Location  indicating value VESSEL_INCONSISTENT_LOCATION, // Inconsistent Location
such as another sea bassin
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          AckCode indicating value InvalidRequestObject,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_02_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a PullRequest with inconsistent Involved agent |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
```

```
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value "Any", // FIXME
              Priority indicating value High,
              Location  indicating value VESSE_LOCATION,
              Agent containing
                identifier indicating value AGENT_UNKNOWN_IDENTIFIER, // Inconsistent Involved
agent
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          AckCode indicating value InvalidRequestObject,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.3 Anomaly service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with timeout on response |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value anomalyService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value anomalyService,
        Payload containing
          Anomaly containing
            AnomalyType indicating value "VesselOutOfTrafficLanes",
            NatureType indicating value "Observed",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
    status_code set to "200 OK",
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value anomalyService,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value anomalyService,
          ServiceOperation indicating value Pull,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
                }
```

### 6.2.1.4 CertificateDocument service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_04" |
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with timeout on response |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value certificateDocumentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value certificateDocumentService,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                 }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BO_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with a vessel document certificate with an invalid Integrity Check |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                 }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value certificateDocumentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Pull,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT_CORRUPTED, // Invalid Integrity Check
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
```

```
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          AckCode indicating value BadRequest,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.5 Incident service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with timeout on response |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
              SickAnimalOnBoard indicating value "true", // Add test for the other features
              UrgencyType indicating value "Immediate",
              SeverityType indicating value "Moderate",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
```

```
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.6 IrregularInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_06" |
|---|---|
| **Test Objective** | Check the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with an IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
```

```
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value irregularMigrationIncidentService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value irregularMigrationIncidentService,
        Payload containing
          IrregularMigrationIncident containing
            identifier indicating value any_value,
            UrgencyType indicating value "Immediate",
            IrregularMigrationIncidentType indicating value "nonSpecified",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
    status_code set to "200 OK",
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value irregularMigrationIncidentService,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value irregularMigrationIncidentService,
          ServiceOperation indicating value Pull,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
               }
```

### 6.2.1.7 LawInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_07" |
| **Test Objective** | Check the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest with a LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value lawInfringementIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value lawInfringementIncidentService,
          Payload containing
            LawInfringementIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
                LawInfringementIncidentType indicating value "nonSpecified",
              Location indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
```

```
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.8 MeteoOceanographicCondition service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest for an ongoing meteo condition |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value meteoOceanographicConditionService,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
```

```
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.9 Organization service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest for an organization |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
```

```
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value organizationService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value organizationService,
        Payload containing
          Organization containing
            Indentifier indicating value any_value,
            OrganizationRole indicating value "portAuthority",
            Location indicating value VESSEL_LOCATION
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
    status_code set to "200 OK",
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value organizationService,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceType indicating value organizationService,
          ServiceOperation indicating value Pull,
        Payload containing
          Organization containing
            AnyField indicating value any_value, // FIXME Which are the fields returned here
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
               }
```

### 6.2.1.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_10" |
| **Test Objective** | Check that the IUT sends a PullResponse before response timeout expiry when receiving a PullRequest for ongoing risk |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value riskService,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
```

```
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value Pull,
          Payload containing
            Organization containing
              AnyField indicating value any_value, // FIXME Which are the fields returned here
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.2.1.11 Cargo service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_REQ_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullResponse before reponse timout expiry when receiving a PullRequest for a cargo |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value Pull,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          PullType indicating value Request,
          DiscoveryProfiles containing
            SeaBasin indicating value SEA_BASIN,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value Pull,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

## 6.2.2 Pull Request multicast

### 6.2.2.1 Vessel service

| **TP Id** | "TP_CDM_NODE_EI_NA_MULTI_PULL_REQ_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving multiple PullRequest for Vessel service - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.3 Multicast Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives several vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
  }
  then {
    the IUT sends multiple HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
  }
                }
```

### 6.2.2.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NA_MULTI_PULL_REQ_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest for Action service - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.3 Multicast Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a action containing
  IMONumber indicating value ACTION_IMO_NUMBER,
  Name indicating value ACTION_NAME,
  Location indicating value ACTION_LOCATION,
```

```
  ShipType indicating value ACTION_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives several vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_nn,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
          Payload containing
              Action containing
                ActionType indicating value "Rescue",
                ActionStatus indicating value "InProgress",
                Mission indicating value nonSpecified,
                Priority indicating value High,
                Location  indicating value VESSEL_LOCATION,
                Agent containing
                  identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
  }
  then {
    the IUT sends multiple HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID_nn,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
        ResultCode indicating value success,
        Signature indicating value any_value
  }
               }
```

## 6.2.3 Pull Discovery

### 6.2.3.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_01" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with Vessel service discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
```

```
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with ActionServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value actionService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
```

```
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.3 Anomaly service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with AnomalyServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value anomalyService,
```

```
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.4 CertificateDocument service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with IncidentServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
```

```
    uri indicating value CDM_PULL_REQUEST_URI,
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        PullType indicating value Discover,
        DiscoveryProfiles containing
          ServiceType indicating value certificateDocumentService,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Payload containing
          vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
            Name indicating value VESSEL_NAME,
            Location indicating value VESSEL_LOCATION,
            ShipType indicating value VESSEL_SHIP_TYPE,
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
            }
```

### 6.2.3.5 Incident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_05" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with IncidentServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value incidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.6 IrregularInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_06" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with IrregularMigrationIncidentService discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value irregularMigrationIncidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
```

```
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.7 LawInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with LawInfringementIncidentService discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value lawInfringementIncidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
```

```
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.8 MeteoOceanographicCondition service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with MeteoOceanographicConditionService discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_METEO_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        PullType indicating value Discover,
        DiscoveryProfiles containing
          ServiceType indicating value meteoOceanographicConditionService,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR and
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vPullResponse containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Payload containing
          vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
            Name indicating value VESSEL_NAME,
            Location indicating value VESSEL_LOCATION,
            ShipType indicating value VESSEL_SHIP_TYPE,
        ResultCode indicating value success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
            }
```

### 6.2.3.9 Organization service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_09" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with OrganizationServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value organizationService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.3.10 Risk service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_10" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with RiskServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value riskService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
```

```
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
            }
```

### 6.2.3.11 Cargo service

| **TP Id** | "TP_CDM_NODE_EI_NA_PULL_DISC_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest with CargoServide discovery - no Acknowledgment required |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover |
| **PICS Selection** | IUT_CDM_NODE and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value cargoService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
```

```
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

## 6.2.4 Push/Subscribe

### 6.2.4.1 Vessel service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for subscription - delegation to the node |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Subscribe,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BO_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response and an error code when receiving a subscription ending in past |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_MINUS_ONE_HOUR,
```

```
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Subscribe,
          AckCode indicating value timestampError, // FIXME To be checked
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.4.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
```

```
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.4.3 Anomaly service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value anomalyService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value anomalyService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
```

```
          ServiceType indicating value anomalyService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value anomalyService,
        AckCode indicating value Success,
        Signature indicating value any_value
    to the ADAPTOR entity
  }
            }
```

### 6.2.4.4 CertificateDocument service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_04" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
```

```
  }
            }
```

| TP Id | "TP_CDM_NODE_EI_NA_SUB_BV_05" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for IncidentService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
             }
```

### 6.2.4.5 Incident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_06" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.4.6 IrregularInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_06" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and<br>CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.4.7 LawInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_07" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and<br>CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
             }
```

### 6.2.4.8 MeteoOceanographicCondition service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_08" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
             }
```

### 6.2.4.9 Organization service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_09" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
             }
```

### 6.2.4.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_10" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.4.11 Cargo service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_BV_11" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a subscription for CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
             }
```

## 6.2.5 Push/Subscribe lookup

### 6.2.5.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_01_01" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a VESSEL_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

| TP Id | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_01_02" |
|---|---|
| Test Objective | Check that the IUT responds with an HTPP 200 OK Acknowledgment response with an empty list when no subscription was done for VesselService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not having a VESSEL_SUBSCRIPTION
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value MESSAGE_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.2 Action service

| TP Id | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_02" |
|---|---|
| Test Objective | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for ActionService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ACTION_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.3 Anomaly service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT having a ANOMALY_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
```

```
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.4 CertificateDocument service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
```

```
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.5 Incident service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.6 IrregularInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_06" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.7 LawInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_07" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a LAW_INFRINGEMENT_INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.8 MeteoOceanographicCondition service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_08" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a METEO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

### 6.2.5.9 Organization service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_09" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ORGANIZATION_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
            }
```

### 6.2.5.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_10" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a RISK_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
            }
```

### 6.2.5.11 Cargo service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_LOOK_BV_11" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request of subscribers list for CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
```

```
the IUT having a CARGO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceCapability containing
              RefreshRate indicating value any_value,
              SubscriptionEnd indicating value END_OF_SUBSCRIPTION,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
              }
```

## 6.2.6 Push/Unsubscribe

### 6.2.6.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_01_01" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a VESSEL_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_01_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response with an error code when receiving a request to delete an unknown subscription for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT not having a VESSEL_SUBSCRIPTION
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value MESSAGE_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value BadRequest, // TODO To be checked
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ACTION_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.3 Anomaly service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_03" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ANOMALY_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
            }
```

### 6.2.6.4 CertificateDocument service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_04" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SUBSCRIPTION containing
```

```
  ContextID indicating value CONTEXT_ID
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
            }
```

### 6.2.6.5 Incident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_05" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a INCIDENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.6 IrregularInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_06" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
```

```
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.7 LawInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a LAW_INFRINGEMENT_INCIDENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.8 MeteoOceanographicCondition service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_08" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a METEO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.9 Organization service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_09" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ORGANIZATION_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

| Expected Behaviour |
|---|
| ensure that {<br>  when {<br>  the IUT receives a vPOST containing<br>  uri indicating value CDM_PULL_REQUEST_URI,<br>  body containing<br>  vPullRequest containing<br>  MessageID indicating value MESSAGE_ID,<br>  CorrelationID indicating value CORRELATION_ID,<br>  ContextID indicating value CONTEXT_ID,<br>  CreationDateTime indicating value CURRENT_TIME,<br>  Priority indicating value PRIORITY,<br>  RequiresAck indicating value false,<br>  Sender containing<br>  ServiceID indicating value SERVICE_ID,<br>  ServiceType indicating value organizationService,<br>  ServiceOperation indicating value subscribe,<br>  Recipient containing<br>  ServiceID indicating value SERVICE_ID,<br>  ServiceType indicating value organizationService,<br>  ServiceOperation indicating value subscribe,<br>  PullType indicating value Unsubscribe,<br>  Signature indicating value any_value<br>  from the ADAPTOR entity<br>  }<br>  then {<br>  the IUT sends a HTTP_RESPONSE containing<br>  status_code set to "200 OK"<br>  body containing<br>  vAcknwoledgment containing<br>  MessageID indicating value MESSAGE_ID,<br>  CorrelationID indicating value CORRELATION_ID,<br>  Priority indicating value PRIORITY,<br>  RequiresAck indicating value false,<br>  AckCode indicating value Success,<br>  Signature indicating value any_value<br>  to the ADAPTOR entity<br>  }<br>  } |

### 6.2.6.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_10" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState and<br>the IUT having a RISK_SERVICE_SUBSCRIPTION containing<br>  ContextID indicating value CONTEXT_ID<br>  } | |
| **Expected Behaviour** | |
| ensure that {<br>  when {<br>  the IUT receives a vPOST containing<br>  uri indicating value CDM_PULL_REQUEST_URI,<br>  body containing<br>  vPullRequest containing<br>  MessageID indicating value MESSAGE_ID,<br>  CorrelationID indicating value CORRELATION_ID,<br>  ContextID indicating value CONTEXT_ID,<br>  CreationDateTime indicating value CURRENT_TIME,<br>  Priority indicating value PRIORITY,<br>  RequiresAck indicating value false,<br>  Sender containing<br>  ServiceID indicating value SERVICE_ID,<br>  ServiceType indicating value riskService,<br>  ServiceOperation indicating value subscribe, | |

```
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
              }
```

### 6.2.6.11 Cargo service

| **TP Id** | "TP_CDM_NODE_EI_NA_SUB_DELETE_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a request to delete subscription for CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CARGO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
```

```
    to the ADAPTOR entity
  }
            }
```

## 6.2.7 Push/Notify

### 6.2.7.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_01" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a VESSEL_SUBSCRIPTION
                }
```

**Expected Behaviour**

```
// FIXME Not sure how it works
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value VesselService,
          ServiceRole indicating value Provider,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
            Name indicating value VESSEL_NAME,
            Location indicating value VESSEL_LOCATION,
            ShipType indicating value VESSEL_SHIP_TYPE,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
                }
```

### 6.2.7.2 Action service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_02" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  ActionType indicating value "Rescue",
  ActionStatus indicating value "InProgress",
  Mission indicating value nonSpecified,
  Priority indicating value High,
  Location indicating value VESSEL_LOCATION and
the IUT having a ACTION_SUBSCRIPTION
           }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value ActionService,
          ServiceRole indicating value Provider,
        Payload containing
          Action containing
            ActionType indicating value "Rescue",
            ActionStatus indicating value "InProgress",
            Mission indicating value nonSpecified,
            Priority indicating value High,
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
           }
```

### 6.2.7.3 Anomaly service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_03" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a ANOMALY_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value AnomalyService,
          ServiceRole indicating value Provider,
        Payload containing
          Anomaly containing
            AnomalyType indicating value "VesselOutOfTrafficLanes",
            NatureType indicating value "Observed",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.4 CertificateDocument service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_04" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a CERTIFICATE_DOCUMENT_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value certificateDocumentService,
          ServiceRole indicating value Provider,
        Payload containing
          Document containing
            Content indicating value RAW_DOCUMENT,
            Hash indicating value SHA_256_RAW_DOCUMENT,
            ReferenceURI  indicating value REFERENCE_URI,
            Location  indicating value VESSE_LOCATION,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.5 Incident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_05" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a INCIDENT_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value incidentService,
          ServiceRole indicating value Provider,
        Payload containing
          Incident containing
            Identifier containing
              GeneratedIn indicating value any_value,
              UUID indicating value any_value,
            SickAnimalOnBoard indicating value "true", // Add test for the other features
            UrgencyType indicating value "Immediate",
            SeverityType indicating value "Moderate",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.6 IrregularInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_06" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value irregularMigrationIncidentService,
          ServiceRole indicating value Provider,
        Payload containing
          IrregularMigrationIncident containing
            identifier indicating value any_value,
            UrgencyType indicating value "Immediate",
            IrregularMigrationIncidentType indicating value "nonSpecified",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.7 LawInfringementIncident service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_07" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value lawInfringementIncidentService,
          ServiceRole indicating value Provider,
        Payload containing
          LawInfringementIncident containing
            identifier indicating value any_value,
            UrgencyType indicating value "Immediate",
              LawInfringementIncidentType indicating value "nonSpecified",
            Location indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.8 MeteoOceanographicCondition service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_08" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a METEO_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value meteoOceanographicConditionService,
          ServiceRole indicating value Provider,
        Payload containing
          MeteoOceanographicCondition containing
            AirTemperature indicating value any_value,
            WindCurrentDirection  indicating value any_value,
            WindCurrentSpeed  indicating value any_value,
            Location indicating value VESSEL_LOCATION,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.9 Organization service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_09" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a ORGANIZATION_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value organizationService,
          ServiceRole indicating value Provider,
        Payload containing
          Organization containing
            Indentifier indicating value any_value,
            OrganizationRole indicating value "portAuthority",
            Location indicating value VESSEL_LOCATION
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_10" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a RISK_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value riskService,
          ServiceRole indicating value Provider,
        Payload containing
          Risk containing
            Indentifier indicating value any_value,
            RiskLevel indicating value high,
            RiskProbability indicating value probable,
            RiskSeverity indicating value critical,
            Location indicating value VESSEL_LOCATION
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

### 6.2.7.11 Cargo service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_PUSH_NOT_BV_11" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Push for notification for CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a CARGO_SUBSCRIPTION
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value cargoService,
          ServiceRole indicating value Provider,
        Payload containing
          Cargo containing
            CargoType indicating value CARGO_TYPE,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the ADAPTOR entity
  }
            }
```

## 6.2.8 Asynchronous Acknowledgement

### 6.2.8.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_ASYNC_ACK_BV_01" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a PullRequest for a Vessel service - Acknowledgment requested |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends multiple HTTP_RESPONSE containing // Asynchronous ACK
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing // Synchronous ACK
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
```

```
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

## 6.2.9 Feedback

### 6.2.9.1 Vessel service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
```

```
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.2 Action Service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a action containing
  IMONumber indicating value ACTION_IMO_NUMBER,
  Name indicating value ACTION_NAME,
  Location indicating value ACTION_LOCATION,
  ShipType indicating value ACTION_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
```

```
          ServiceType indicating value actionService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Payload containing
          Action containing
            ActionType indicating value "Rescue",
            ActionStatus indicating value "InProgress",
            Mission indicating value nonSpecified,
            Priority indicating value High,
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Signature indicating value any_value,
        FeedbackType indicating value Info,
        Reason indicating value ShipTypeChange,
        RefMessageID indicating value REF_MESSAGE_ID
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        AckCode indicating value Success,
        Signature indicating value any_value
  to the ADAPTOR entity
}
             }
```

### 6.2.9.3 Anomaly Service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a anomaly containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value anomalyService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
```

```
          ServiceOperation indicating value Pull,
        Payload containing
          Anomaly containing
            AnomalyType indicating value "VesselOutOfTrafficLanes",
            NatureType indicating value "Observed",
            Location  indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        Signature indicating value any_value,
        FeedbackType indicating value Info,
        Reason indicating value ShipTypeChange,
        RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.4 Certificate Document Service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_04" |
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a certificateDocument containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value certificateDocumentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
```

```
            Location  indicating value VESSE_LOCATION,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

### 6.2.9.5 Incident Service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a Incident containing
  Identifier containing
    GeneratedIn indicating value any_value,
    UUID indicating value any_value,
    SickAnimalOnBoard indicating value "true", // Add test for the other features
    UrgencyType indicating value "Immediate",
    SeverityType indicating value "Moderate",
    Location  indicating value VESSEL_LOCATION,
    Agent containing
      identifier indicating value AGENT_IDENTIFIER and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
                SickAnimalOnBoard indicating value "true", // Add test for the other features
```

```
                UrgencyType indicating value "Immediate",
                SeverityType indicating value "Moderate",
                Location  indicating value VESSEL_LOCATION,
                Agent containing
                  identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.6 IrregularMigrationIncident Service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_06" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for IrregularMigrationIncident |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and<br>CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a irregularMigrationIncident containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value irregularMigrationIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Incident containing
              IrregularMigrationIncident containing
                identifier indicating value any_value,
                UrgencyType indicating value "Immediate",
                IrregularMigrationIncidentType indicating value "nonSpecified",
```

```
                Location  indicating value VESSEL_LOCATION,
                Agent containing
                  identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.7 LawInfringimentIncident Service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and<br>CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a LawInfringementIncident containing
  identifier indicating value any_value,
  UrgencyType indicating value "Immediate",
    LawInfringementIncidentType indicating value "nonSpecified",
  Location indicating value VESSEL_LOCATION,
  Agent containing
    identifier indicating value AGENT_IDENTIFIER and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value lawInfringementIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            LawInfringementIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
                LawInfringementIncidentType indicating value "nonSpecified",
              Location indicating value VESSEL_LOCATION,
```

```
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.8 MeteoOceanographicCondition service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a meteoOceanographicConditionService containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
```

```
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.9 Organization service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a organization containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value organizationService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Organization containing
              Indentifier indicating value any_value,
              OrganizationRole indicating value "portAuthority",
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
```

```
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.10 Risk service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_10" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a risk containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
```

```
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

### 6.2.9.11 Cargo service

| **TP Id** | "TP_CDM_NODE_EI_NA_FEEDBACK_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response when receiving a Feedback for CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Cargo Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a cargo containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value cargoService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                }
```

# 6.3 EI NN Interface

## 6.3.1 Pull Request

### 6.3.1.1 Vessel service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_01_01" |
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for a VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                 }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
```

```
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_01_02" |
|---|---|
| **Test Objective** | Check that the IUT sends the synchronous Acknowledgment to the Adaptor when receiving the synchronous Acknowledgment from the GatewayProvider |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
```

```
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER and
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_01_03" |
|---|---|
| **Test Objective** | Check that the IUT sends the PullResponse to the Adaptor when receiving the PullResponse from the GatewayProvider |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
      the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
        status_code set to "200 OK",
```

```
       body containing
         vPullResponse containing
           MessageID indicating value MESSAGE_ID,
           CorrelationID indicating value CORRELATION_ID,
           CreationDateTime indicating value CURRENT_TIME,
           Priority indicating value PRIORITY,
           RequiresAck indicating value false,
           Sender containing
             ServiceID indicating value SERVICE_ID,
             ServiceOperation indicating value Pull,
           ServiceType indicating value vesselService,
           Recipient containing
             ServiceID indicating value SERVICE_ID,
             ServiceOperation indicating value Pull,
           ServiceType indicating value vesselService,
           Payload containing
             vessel containing
               IMONumber indicating value VESSEL_IMO_NUMBER,
               Name indicating value VESSEL_NAME,
               Location indicating value VESSEL_LOCATION,
               ShipType indicating value VESSEL_SHIP_TYPE,
           ResultCode indicating value success,
           Signature indicating value any_value
     from the GATEWAY_PROVIDER entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
        status_code set to "200 OK",
        body containing
          vPullResponse containing
            MessageID indicating value MESSAGE_ID,
            CorrelationID indicating value CORRELATION_ID,
            CreationDateTime indicating value CURRENT_TIME,
            Priority indicating value PRIORITY,
            RequiresAck indicating value false,
            Sender containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull,
            Recipient containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull,
            Payload containing
              vessel containing
                IMONumber indicating value VESSEL_IMO_NUMBER,
                Name indicating value VESSEL_NAME,
                Location indicating value VESSEL_LOCATION,
                ShipType indicating value VESSEL_SHIP_TYPE,
            ResultCode indicating value success,
            Signature indicating value any_value
    to the ADAPTOR entity
  }
                  }
```

### 6.3.1.2 Action service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Action Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a action containing
  IMONumber indicating value ACTION_IMO_NUMBER,
  Name indicating value ACTION_NAME,
  Location indicating value ACTION_LOCATION,
  ShipType indicating value ACTION_SHIP_TYPE
                  }
```

| **Expected Behaviour** |
|---|

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value Pull,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value nonSpecified,
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value actionService,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value nonSpecified,
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                }
```

### 6.3.1.3 Anomaly service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_03" |
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for am AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Anomaly Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a anomaly containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Payload containing
            Anomaly containing
              AnomalyType indicating value "VesselOutOfTrafficLanes",
              NatureType indicating value "Observed",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value Pull,
          Payload containing
            Anomaly containing
              AnomalyType indicating value "VesselOutOfTrafficLanes",
              NatureType indicating value "Observed",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
```

```
            identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
              }
```

### 6.3.1.4 CertificateDocument service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_04" |
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for a CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a anomaly containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Recipient containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value anomalyService,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                  }
```

### 6.3.1.5 Incident service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a incident containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
              SickAnimalOnBoard indicating value "true", // Add test for the other features
              UrgencyType indicating value "Immediate",
              SeverityType indicating value "Moderate",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
```

```
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value incidentService,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
              SickAnimalOnBoard indicating value "true", // Add test for the other features
              UrgencyType indicating value "Immediate",
              SeverityType indicating value "Moderate",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                  }
```

### 6.3.1.6 IrregularMigrationIncident service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_06" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an IrregularMigrationIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a irregularMigrationIncident containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value irregularMigrationIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
```

```
            ServiceOperation indicating value Pull,
            ServiceType indicating value irregularMigrationIncidentService,
          Payload containing
            IrregularMigrationIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
              IrregularMigrationIncidentType indicating value "nonSpecified",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value irregularMigrationIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value irregularMigrationIncidentService,
          Payload containing
            IrregularMigrationIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
              IrregularMigrationIncidentType indicating value "nonSpecified",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                }
```

### 6.3.1.7 LawInfringementIncident service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an LawInfringementIncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a lawInfringementIncident containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value lawInfringementIncidentService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value lawInfringementIncidentService,
        Payload containing
          LawInfringementIncident containing
            identifier indicating value any_value,
            UrgencyType indicating value "Immediate",
              LawInfringementIncidentType indicating value "nonSpecified",
            Location indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a vPOST containing
    uri indicating value CDM_PULL_REQUEST_URI,
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID_1,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME_1,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value lawInfringementIncidentService,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
          ServiceType indicating value lawInfringementIncidentService,
        Payload containing
          LawInfringementIncident containing
            identifier indicating value any_value,
            UrgencyType indicating value "Immediate",
              LawInfringementIncidentType indicating value "nonSpecified",
            Location indicating value VESSEL_LOCATION,
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  to the GATEWAY_PROVIDER entity
}
               }
```

### 6.3.1.8 MeteoOceanographicCondition service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_08" |
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an ongoing meteo condition |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a meteoOceanographicCondition containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value Pull,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value Pull,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
```

```
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                }
```

### 6.3.1.9 Organization service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for an OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a organization containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value organizationService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value Pull,
          Payload containing
            Organization containing
              Indentifier indicating value any_value,
              OrganizationRole indicating value "portAuthority",
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value organizationService,
```

```
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value organizationService,
          Payload containing
            Organization containing
              Indentifier indicating value any_value,
              OrganizationRole indicating value "portAuthority",
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                  }
```

### 6.3.1.10 Risk service

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_10" |
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for a RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a risk containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value Pull,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
```

```
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value riskService,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                  }
```

### 6.3.1.11 Cargo service

| **TP Id** | "TP_CDM_NODE_EI_NN_PULL_REQ_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for a CargoService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not havingRegistered a GATEWAY_PROVIDER and
the IUT not havingRegistered a cargo containing
  IMONumber indicating value ANOMALY_IMO_NUMBER,
  Name indicating value ANOMALY_NAME,
  Location indicating value ANOMALY_LOCATION,
  ShipType indicating value ANOMALY_SHIP_TYPE
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value cargoService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value cargoService,
          Payload containing
            Cargo containing
```

```
            CargoType indicating value CARGO_TYPE,
        PullType indicating value Request,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID_1,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME_1,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value cargoService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value cargoService,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                }
```

## 6.3.2 Pull Discovery

| **TP Id** | "TP_CDM_NODE_EI_NN_MULTI_PULL_REQ_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest to the GatewayProvider when receiving a PullRequest for a MeteoOceanographicConditionService discovery |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
```

```
          ServiceType indicating value meteoOceanographicConditionService,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  from the ADAPTOR entity
}
then {
  the IUT sends a vPOST containing
    uri indicating value CDM_PULL_REQUEST_URI,
    body containing
      vPullRequest containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        PullType indicating value Discover,
        DiscoveryProfiles containing
          ServiceType indicating value meteoOceanographicConditionService,
        ResponseTimeOut indicating value RESPONSE_TIMEOUT,
        Signature indicating value any_value
  to the GATEWAY_PROVIDER entity
}
                }
```

## 6.3.3 Push/Subscribe

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_SUB_BV_01" |
| **Test Objective** | Check that the IUT sends a PullRequest for subscription to the GatewayProvider when receiving a PullRequest for a VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
```

```
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                  }
```

## 6.3.4 Push/Subscribe lookup

| **TP Id** | "TP_CDM_NODE_EI_NN_SUB_LOOK_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for subscribers list to the GatewayProvider when receiving a PullRequest for subscribers list for a IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
                  }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the GATEWAY_PROVIDER entity
  }
                }
```

## 6.3.5 Push/Unsubscribe

| **TP Id** | "TP_CDM_NODE_EI_NN_SUB_DELETE_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for subscription deletion to the GatewayProvider when receiving a PullRequest for subscription deletion for a RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a RISK_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    from the GATEWAY_PROVIDER entity
```

```
  }
                }
```

## 6.3.6 Push/Notify

| | |
|---|---|
| **TP Id** | "TP_CDM_NODE_EI_NN_PUSH_NOT_BV_01" |
| **Test Objective** | Check that the IUT sends a Push to the Adaptor when receiving a Push from the GatewayProvider for a VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT having a VESSEL_SUBSCRIPTION
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value VesselService,
          ServiceRole indicating value Provider,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
            Name indicating value VESSEL_NAME,
            Location indicating value VESSEL_LOCATION,
            ShipType indicating value VESSEL_SHIP_TYPE,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the GATEWAY_PROVIDER entity
  }
  then {
    the ADAPTOR sends a vPOST containing
    uri indicating value CDM_PUSH_REQUEST_URI,
    body containing
      vPush containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        CreationDateTime indicating value CURRENT_TIME,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value subscribe,
          ServiceType indicating value VesselService,
          ServiceRole indicating value Provider,
        Payload containing
          Vessel containing
            IMONumber indicating value VESSEL_IMO_NUMBER,
            Name indicating value VESSEL_NAME,
            Location indicating value VESSEL_LOCATION,
            ShipType indicating value VESSEL_SHIP_TYPE,
        Reliability  indicating value HighReliability,
        Signature indicating value any_value
    from the ADAPTOR entity
```

```
  }
                }
```

## 6.3.7 Asynchronous Acknowledgement

| TP Id | "TP_CDM_NODE_EI_NN_ASYNC_ACK_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for async Ack to the GatewayProvider when receiving a PullRequest for async Ack for a RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
              Cargo containing
                CargoType indicating value CARGO_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the ADAPTOR entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
              Cargo containing
                CargoType indicating value CARGO_TYPE,
          PullType indicating value Request,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    from the GATEWAY_PROVIDER and
    the IUT sends multiple HTTP_RESPONSE containing // Asynchronous ACK
      status_code set to "200 OK"
```

```
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sends a HTTP_RESPONSE containing // Synchronous ACK
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR and
    the IUT sendsBeforeResponseTimerExpiry a HTTP_RESPONSE containing
      status_code set to "200 OK",
      body containing
        vPullResponse containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          ResultCode indicating value success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
               }
```

## 6.3.8 Feedback

| **TP Id** | "TP_CDM_NODE_EI_NN_FEEDBACK_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for feedback to the GatewayProvider when receiving a PullRequest for feedback for a VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState and
the IUT havingRegistered a vessel containing
  IMONumber indicating value VESSEL_IMO_NUMBER,
```

```
  Name indicating value VESSEL_NAME,
  Location indicating value VESSEL_LOCATION,
  ShipType indicating value VESSEL_SHIP_TYPE and
the IUT havingReceived a PullRequest containing
  MessageID indicating value REF_MESSAGE_ID
               }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    from the GATEWAY_PROVIDER entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value vesselService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the ADAPTOR entity
  }
               }
```

# 6.4 Adaptor

## 6.4.1 Pull Request

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_01" |
| **Test Objective** | Check that the IUT sends a PullRequest for VesselService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>} | |
| **Expected Behaviour** | |
| ensure that {<br>when {<br>the IUT isTriggeredToSend a PullRequest containing<br>PullType indicating value Request,<br>Vessel containing<br>IMONumber indicating value VESSEL_IMO_NUMBER,<br>Name indicating value VESSEL_NAME,<br>Location indicating value VESSEL_LOCATION,<br>ShipType indicating value VESSEL_SHIP_TYPE<br>from the TEST_SYSTEM entity<br>}<br>then {<br>the IUT sends a HTTP_REQUEST containing<br>uri indicating value CDM_PULL_REQUEST_URI,<br>body containing<br>vPullRequest containing<br>MessageID indicating value MESSAGE_ID,<br>CorrelationID indicating value CORRELATION_ID,<br>CreationDateTime indicating value CURRENT_TIME,<br>Priority indicating value PRIORITY,<br>RequiresAck indicating value false,<br>Sender containing<br>ServiceID indicating value SERVICE_ID,<br>ServiceOperation indicating value Pull,<br>Recipient containing<br>ServiceID indicating value SERVICE_ID,<br>ServiceOperation indicating value Pull,<br>Payload containing<br>Vessel containing<br>IMONumber indicating value VESSEL_IMO_NUMBER,<br>Name indicating value VESSEL_NAME,<br>Location indicating value VESSEL_LOCATION,<br>ShipType indicating value VESSEL_SHIP_TYPE,<br>PullType indicating value Request,<br>Signature indicating value any_value<br>to the NODE entity<br>}<br>} | |

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_02" |
| **Test Objective** | Check that the IUT sends a PullRequest for ActionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ACTION_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>} | |
| **Expected Behaviour** | |
| ensure that {<br>when { | |

```
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Action containing
        ActionType indicating value "Rescue",
        ActionStatus indicating value "InProgress",
        Mission indicating value nonSpecified,
        Priority indicating value High,
        Location  indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
              Mission indicating value nonSpecified,
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for AnomalyService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ANOMALY_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Anomaly containing
        AnomalyType indicating value "VesselOutOfTrafficLanes",
        NatureType indicating value "Observed",
        Location  indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Anomaly containing
              AnomalyType indicating value "VesselOutOfTrafficLanes",
              NatureType indicating value "Observed",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                 identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for CertificateDocumentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Document containing
        Content indicating value RAW_DOCUMENT,
        Hash indicating value SHA_256_RAW_DOCUMENT,
        ReferenceURI  indicating value REFERENCE_URI,
        Location  indicating value VESSE_LOCATION
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_05" |
| **Test Objective** | Check that the IUT sends a PullRequest for IncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Incident containing
        Identifier containing
          GeneratedIn indicating value any_value,
          UUID indicating value any_value,
        SickAnimalOnBoard indicating value "true", // Add test for the other features
        UrgencyType indicating value "Immediate",
        SeverityType indicating value "Moderate",
        Location  indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
              SickAnimalOnBoard indicating value "true", // Add test for the other features
              UrgencyType indicating value "Immediate",
              SeverityType indicating value "Moderate",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_REQ_BV_06" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for IrregularMigrationIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and<br>CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      IrregularMigrationIncident containing
        identifier indicating value any_value,
        UrgencyType indicating value "Immediate",
        IrregularMigrationIncidentType indicating value "nonSpecified",
        Location  indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            IrregularMigrationIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
              IrregularMigrationIncidentType indicating value "nonSpecified",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_REQ_BV_07" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for LawInfringementIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
```

```
      PullType indicating value Request,
      LawInfringementIncident containing
        identifier indicating value any_value,
        UrgencyType indicating value "Immediate",
          LawInfringementIncidentType indicating value "nonSpecified",
        Location indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            LawInfringementIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
                LawInfringementIncidentType indicating value "nonSpecified",
              Location indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for MeteoOceanographicConditionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_METEO_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      MeteoOceanographicCondition containing
        AirTemperature indicating value any_value,
        WindCurrentDirection  indicating value any_value,
        WindCurrentSpeed  indicating value any_value,
        Location indicating value VESSEL_LOCATION
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
```

```
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_REQ_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for OrganizationService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Organization containing
        Indentifier indicating value any_value,
        OrganizationRole indicating value "portAuthority",
        Location indicating value VESSEL_LOCATION
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Organization containing
              Indentifier indicating value any_value,
              OrganizationRole indicating value "portAuthority",
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_REQ_BV_10" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for RiskService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Request,
      Risk containing
        Indentifier indicating value any_value,
        RiskLevel indicating value high,
        RiskProbability indicating value probable,
        RiskSeverity indicating value critical,
        Location indicating value VESSEL_LOCATION
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_REQ_BV_11" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for CargoService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
```

```
      PullType indicating value Request,
      Cargo containing
        CargoType indicating value CARGO_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          PullType indicating value Request,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

## 6.4.2 Pull Discovery

| **TP Id** | "TP_CDM_ADAPTOR_PULL_DISC_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for VesselService discovery when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value vesselService,
        IMONumber indicating value VESSEL_IMO_NUMBER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value vesselService,
            IMONumber indicating value VESSEL_IMO_NUMBER,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
```

```
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_02" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for ActionService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value actionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value actionService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_03" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for AnomalyService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value anomalyService
    from the TEST_SYSTEM entity
```

```
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value anomalyService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_DISC_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for CertificateDocumentService discovery when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value certificateDocumentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value certificateDocumentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_05" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for IncidentService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value incidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value incidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_06" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for IrregularMigrationIncidentService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value irregularMigrationIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
```

```
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value irregularMigrationIncidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_DISC_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for LawInfringementIncidentService discovery when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value lawInfringementIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value lawInfringementIncidentService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_08" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for MeteoOceanographicConditionService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value meteoOceanographicConditionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value meteoOceanographicConditionService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_09" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for OrganizationService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value organizationService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
```

```
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value organizationService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
      to the NODE entity
    }
                }
```

| **TP Id** | "TP_CDM_ADAPTOR_PULL_DISC_BV_10" |
|---|---|
| **Test Objective** | Check that the IUT sends a PullRequest for RiskService discovery when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause .1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_ADAPTOR and CDM_PULL and CDM_RISK_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
                }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Discover,
      DiscoveryProfiles containing
        ServiceType indicating value riskService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a HTTP_REQUEST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          PullType indicating value Discover,
          DiscoveryProfiles containing
            ServiceType indicating value riskService,
          ResponseTimeOut indicating value RESPONSE_TIMEOUT,
          Signature indicating value any_value
    to the NODE entity
  }
                }
```

| TP Id | "TP_CDM_ADAPTOR_PULL_DISC_BV_11" |
|---|---|
| Test Objective | Check that the IUT sends a PullRequest for CargoService discovery when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 5.5 CISE Discovery mechanism<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 004 [1], clause 8.1.7 Discover<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_ADAPTOR and CDM_PULL and CDM_CARGO_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>} | |
| **Expected Behaviour** | |
| ensure that {<br>when {<br>the IUT isTriggeredToSend a PullRequest containing<br>PullType indicating value Discover,<br>DiscoveryProfiles containing<br>ServiceType indicating value cargoService<br>from the TEST_SYSTEM entity<br>}<br>then {<br>the IUT sends a HTTP_REQUEST containing<br>uri indicating value CDM_PULL_REQUEST_URI,<br>body containing<br>vPullRequest containing<br>MessageID indicating value MESSAGE_ID,<br>CorrelationID indicating value CORRELATION_ID,<br>CreationDateTime indicating value CURRENT_TIME,<br>Priority indicating value PRIORITY,<br>RequiresAck indicating value false,<br>Sender containing<br>ServiceID indicating value SERVICE_ID,<br>ServiceOperation indicating value Pull,<br>PullType indicating value Discover,<br>DiscoveryProfiles containing<br>ServiceType indicating value cargoService,<br>ResponseTimeOut indicating value RESPONSE_TIMEOUT,<br>Signature indicating value any_value<br>to the NODE entity<br>}<br>} | |

## 6.4.3 Push/Subscribe

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_01" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for VesselService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>} | |
| **Expected Behaviour** | |
| ensure that {<br>when {<br>the IUT isTriggeredToSend a PullRequest containing<br>PullType indicating value Subscribe,<br>SeaBasin indicating value SEA_BASIN,<br>Requests containing<br>ServiceType indicating value vesselService,<br>SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR<br>from the TEST_SYSTEM entity<br>}<br>then {<br>the IUT sends a vPOST containing<br>uri indicating value CDM_PULL_REQUEST_URI,<br>body containing<br>vPullRequest containing | |

```
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value vesselService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT sends a subscription for ActionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ACTION_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value actionService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value actionService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_03" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for AnomalyService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value anomalyService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value anomalyService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value anomalyService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_04" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for CertificateDocumentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value certificateDocumentService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
```

```
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value certificateDocumentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT sends a subscription for IncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_INCIDENT_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value incidentService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value incidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
```

```
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_06" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for IrregularMigrationIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and<br>CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value irregularMigrationIncidentService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value irregularMigrationIncidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_07" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for LawInfringementIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and<br>CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
```

```
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value lawInfringementIncidentService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value lawInfringementIncidentService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT sends a subscription for MeteoOceanographicConditionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value meteoOceanographicConditionService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
            SeaBasin indicating value SEA_BASIN,
```

```
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value meteoOceanographicConditionService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT sends a subscription for OrganizationService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value organizationService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value organizationService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_10" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for RiskService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value riskService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value riskService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_BV_11" |
|---|---|
| Test Objective | Check that the IUT sends a subscription for CargoService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_SUBSCRIBE and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Subscribe,
      SeaBasin indicating value SEA_BASIN,
      Requests containing
        ServiceType indicating value cargoService,
        SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR
    from the TEST_SYSTEM entity
  }
```

```
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
            SeaBasin indicating value SEA_BASIN,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value cargoService,
          ResponseTimeOut indicating value 1000,
          PullType indicating value Subscribe,
          Requests containing
            SubscriptionEnd indicating value CURENT_TIME_PLUS_ONE_HOUR,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

## 6.4.4 Push/Subscribe lookup

| **TP Id** | "TP_CDM_ADAPTOR_SUB_LOOK_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT sends a request of subscribers list for IncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a VESSEL_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value vesselService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_02" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for ActionService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having an ACTION_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value actionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_03" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for AnomalyService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having an ANOMALY_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value anomalyService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
```

```
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_LOOK_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT sends a request of subscribers list for CertificateDocumentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value certificateDocumentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value certificateDocumentService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_05" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for IncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having an INCIDENT_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value incidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_06" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for IrregularMigrationIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value irregularMigrationIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
```

```
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_LOOK_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT sends a request of subscribers list for LawInfringementIncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a LAW_INFRINGEMENT_INCIDENT_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value lawInfringementIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_08" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for MeteoOceanographicConditionService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a METEO_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value meteoOceanographicConditionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_09" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for OrganizationService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.15.2.1 Organization Class (subclass of Agent)<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ORGANIZATION_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value organizationService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
```

```
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_LOOK_BV_10" |
|---|---|
| **Test Objective** | Check that the IUT sends a request of subscribers list for RiskService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a RISK_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value riskService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_LOOK_BV_11" |
|---|---|
| Test Objective | Check that the IUT sends a request of subscribers list for CargoService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.8 Get Subscribers<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CARGO_SERVICE_SUBSCRIPTION containing
   ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value GetSubscribers,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value cargoService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT sends a vPOST containing
      uri indicating value URI_SUBSCRIBE_GET,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value Subscribe,
          PullType indicating value GetSubscribers,
          ResponseTimeOut indicating value 1000,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

## 6.4.5 Push/Unsubscribe

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_01" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for RiskService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a VESSEL_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value vesselService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value vesselService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_DELETE_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT sends a request to delete subscription for ActionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ACTION_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value actionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value actionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_03" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for AnomalyService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ANOMALY_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value anomalyService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value anomalyService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_04" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for CertificateDocumentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value certificateDocumentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
```

```
       vPullRequest containing
         MessageID indicating value MESSAGE_ID,
         CorrelationID indicating value CORRELATION_ID,
         ContextID indicating value CONTEXT_ID,
         CreationDateTime indicating value CURRENT_TIME,
         Priority indicating value PRIORITY,
         RequiresAck indicating value false,
         Sender containing
           ServiceID indicating value SERVICE_ID,
           ServiceType indicating value certificateDocumentService,
           ServiceOperation indicating value subscribe,
         Recipient containing
           ServiceID indicating value SERVICE_ID,
           ServiceType indicating value certificateDocumentService,
           ServiceOperation indicating value subscribe,
         PullType indicating value Unsubscribe,
         Signature indicating value any_value
    to the NODE entity
  }
             }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_DELETE_BV_05" |
|---|---|
| **Test Objective** | Check that the IUT sends a request to delete subscription for IncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value incidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value incidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
             }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_06" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for IrregularMigrationIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value irregularMigrationIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value irregularMigrationIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_07" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for LawInfringementIncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a LAW_INFRINGEMENT_INCIDENT_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value lawInfringementIncidentService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
```

```
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value lawInfringementIncidentService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_DELETE_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT sends a request to delete subscription for MeteoOceanographicConditionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a METEO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value meteoOceanographicConditionService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value meteoOceanographicConditionService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_09" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for OrganizationService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ORGANIZATION_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value organizationService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value organizationService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_SUB_DELETE_BV_10" |
|---|---|
| Test Objective | Check that the IUT sends a request to delete subscription for RiskService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| PICS Selection | IUT_CDM_NODE and CDM_PULL and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a RISK_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value riskService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
```

```
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value riskService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_SUB_DELETE_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT sends a request to delete subscription for CargoService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CARGO_SERVICE_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a PullRequest containing
      PullType indicating value Unsubscribe,
      Requests containing
        ContextID indicating value CONTEXT_ID,
        ServiceType indicating value cargoService
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          ContextID indicating value CONTEXT_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value subscribe,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceType indicating value cargoService,
            ServiceOperation indicating value subscribe,
          PullType indicating value Unsubscribe,
          Signature indicating value any_value
    to the NODE entity
  }
              }
```

## 6.4.6 Push/Notify

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_01_01" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment with an error code response when receiving a Push for notification for an unknown subscription for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a VESSEL_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
             }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value VesselService,
            ServiceRole indicating value Provider,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_01_02" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment with an error code response when receiving a Push for notification for an unknown subscription for VesselService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT not having a VESSEL_SUBSCRIPTION
             }
```

**Expected Behaviour**

```
ensure that {
  when {
```

```
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value VesselService,
            ServiceRole indicating value Provider,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
              ShipType indicating value VESSEL_SHIP_TYPE,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
      from the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value badRequest
    to the NODE entity
  }
}
```

| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_02" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for ActionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ACTION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ACTION_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value ActionService,
            ServiceRole indicating value Provider,
          Payload containing
            Action containing
              ActionType indicating value "Rescue",
              ActionStatus indicating value "InProgress",
```

```
              Mission indicating value nonSpecified,
              Priority indicating value High,
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
            Reliability  indicating value HighReliability,
            Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for AnomalyService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ANOMALY_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value AnomalyService,
            ServiceRole indicating value Provider,
          Payload containing
            Anomaly containing
              AnomalyType indicating value "VesselOutOfTrafficLanes",
              NatureType indicating value "Observed",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
            Reliability  indicating value HighReliability,
            Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
```

```
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for CertificateDocumentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CERTIFICATE_DOCUMENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value CertificateDocumentService,
            ServiceRole indicating value Provider,
          Payload containing
            Document containing
              Content indicating value RAW_DOCUMENT,
              Hash indicating value SHA_256_RAW_DOCUMENT,
              ReferenceURI  indicating value REFERENCE_URI,
              Location  indicating value VESSE_LOCATION,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_05" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for IncidentService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a INCIDENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value IncidentService,
            ServiceRole indicating value Provider,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
              SickAnimalOnBoard indicating value "true", // Add test for the other features
              UrgencyType indicating value "Immediate",
              SeverityType indicating value "Moderate",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| TP Id | "TP_CDM_ADAPTOR_PUSH_NOT_BV_06" |
|---|---|
| Test Objective | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for IrregularMigrationIncidentService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PUSH and CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a IRREGULAR_MIGRATION_INCIDENT_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value IrregularMigrationIncidentService,
            ServiceRole indicating value Provider,
          Payload containing
            IrregularMigrationIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
              IrregularMigrationIncidentType indicating value "nonSpecified",
              Location  indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| TP Id | "TP_CDM_ADAPTOR_PUSH_NOT_BV_07" |
|---|---|
| Test Objective | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for LawInfringementIncidentService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PUSH and CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a LAW_INFRINGEMENT_INCIDENT_SUBSCRIPTION containing
```

```
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value LawInfringementIncidentService,
            ServiceRole indicating value Provider,
          Payload containing
            LawInfringementIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
                LawInfringementIncidentType indicating value "nonSpecified",
              Location indicating value VESSEL_LOCATION,
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_08" |
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for MeteoOceanographicConditionService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a METEO_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value MeteoOceanographicConditionService,
            ServiceRole indicating value Provider,
          Payload containing
            MeteoOceanographicCondition containing
              AirTemperature indicating value any_value,
              WindCurrentDirection  indicating value any_value,
              WindCurrentSpeed  indicating value any_value,
              Location indicating value VESSEL_LOCATION,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_09" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for OrganizationService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a ORGANIZATION_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value OrganizationService,
            ServiceRole indicating value Provider,
          Payload containing
            Organization containing
              Indentifier indicating value any_value,
              OrganizationRole indicating value "portAuthority",
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
```

```
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| **TP Id** | "TP_CDM_ADAPTOR_PUSH_NOT_BV_10" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for RiskService |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_PUSH and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a RISK_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value RiskService,
            ServiceRole indicating value Provider,
          Payload containing
            Risk containing
              Indentifier indicating value any_value,
              RiskLevel indicating value high,
              RiskProbability indicating value probable,
              RiskSeverity indicating value critical,
              Location indicating value VESSEL_LOCATION
              Agent containing
                identifier indicating value AGENT_IDENTIFIER,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

| TP Id | "TP_CDM_ADAPTOR_PUSH_NOT_BV_11" |
|---|---|
| Test Objective | Check that the IUT responds with an HTPP 200 OK Acknowledgment response when receiving a Push for notification for CargoService |
| Reference | ETSI GS CDM 004 [1], clause 5.4.6 Publish/Subscribe<br>ETSI GS CDM 004 [1], clause 8.1.2 Push<br>ETSI GS CDM 004 [1], clause 8.1.6 Publish/Subscribe<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_PUSH and CDM_CARGO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState and
the IUT having a CARGO_SUBSCRIPTION containing
  ContextID indicating value CONTEXT_ID
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PUSH_REQUEST_URI,
      body containing
        vPush containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value subscribe,
            ServiceType indicating value CargoService,
            ServiceRole indicating value Provider,
          Payload containing
            Cargo containing
              CargoType indicating value CARGO_TYPE,
          Reliability  indicating value HighReliability,
          Signature indicating value any_value
    to the NODE entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          AckCode indicating value Success,
          DiscoveredServices containing
            ServiceID indicating value SERVICE_ID_CONSUMER
    to the NODE entity
  }
}
```

## 6.4.7 Asynchronous Acknowledgement

Void.

## 6.4.8 Feedback

| TP Id | "TP_CDM_ADAPTOR_FEEDBACK_BV_01" |
|---|---|
| Test Objective | Check that the IUT sends a Feedback for VesselService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.20 Vessel Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_FEEDBACK and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

| Expected Behaviour |
|---|
| ensure that {<br>  when {<br>  the IUT isTriggeredToSend a Feedback containing<br>  Vessel containing<br>  IMONumber indicating value VESSEL_IMO_NUMBER,<br>  Name indicating value VESSEL_NAME,<br>  Location indicating value VESSEL_LOCATION,<br>  ShipType indicating value VESSEL_SHIP_TYPE<br>  from the TEST_SYSTEM entity<br>  }<br>  then {<br>  the IUT receives a vPOST containing<br>  uri indicating value CDM_FEEDBACK_REQUEST_URI,<br>  body containing<br>  vFeedback containing<br>  MessageID indicating value MESSAGE_ID,<br>  CorrelationID indicating value CORRELATION_ID,<br>  CreationDateTime indicating value CURRENT_TIME,<br>  Priority indicating value PRIORITY,<br>  RequiresAck indicating value true,<br>  Sender containing<br>  ServiceID indicating value SERVICE_ID,<br>  ServiceOperation indicating value Feedback,<br>  ServiceType indicating value vesselService,<br>  Recipient containing<br>  ServiceID indicating value SERVICE_ID,<br>  ServiceOperation indicating value Pull,<br>  Payload containing<br>  Vessel containing<br>  IMONumber indicating value VESSEL_IMO_NUMBER,<br>  Name indicating value VESSEL_NAME,<br>  Location indicating value VESSEL_LOCATION,<br>  ShipType indicating value VESSEL_SHIP_TYPE,<br>  Signature indicating value any_value,<br>  FeedbackType indicating value Info,<br>  Reason indicating value ShipTypeChange,<br>  RefMessageID indicating value REF_MESSAGE_ID<br>  to the NODE entity<br>  }<br>  } |

| | |
|---|---|
| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_02" |
| **Test Objective** | Check that the IUT sends a Feedback for ActionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.3 Action Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.10 Location Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ACTION_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>  } | |
| **Expected Behaviour** | |
| ensure that {<br>  when {<br>  the IUT isTriggeredToSend a Feedback containing<br>  Action containing<br>  IMONumber indicating value ACTION_IMO_NUMBER,<br>  Name indicating value ACTION_NAME,<br>  Location indicating value ACTION_LOCATION,<br>  ShipType indicating value ACTION_SHIP_TYPE<br>  from the TEST_SYSTEM entity<br>  }<br>  then {<br>  the IUT receives a vPOST containing<br>  uri indicating value CDM_FEEDBACK_REQUEST_URI,<br>  body containing<br>  vFeedback containing<br>  MessageID indicating value MESSAGE_ID,<br>  CorrelationID indicating value CORRELATION_ID,<br>  CreationDateTime indicating value CURRENT_TIME,<br>  Priority indicating value PRIORITY, | |

```
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value actionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Action containing
              IMONumber indicating value ACTION_IMO_NUMBER,
              Name indicating value ACTION_NAME,
              Location indicating value ACTION_LOCATION,
              ShipType indicating value ACTION_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_03" |
|---|---|
| **Test Objective** | Check that the IUT sends a Feedback for AnomalyService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.5 Anomaly Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ANOMALY_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      Anomaly containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value anomalyService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Anomaly containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_FEEDBACK_BV_04" |
|---|---|
| Test Objective | Check that the IUT sends a Feedback for CertificateDocumentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.7 Document Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_FEEDBACK and CDM_CERTIFICATE_DOCUMENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      certificateDocument containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value certificateDocumentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            certificateDocument containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| TP Id | "TP_CDM_ADAPTOR_FEEDBACK_BV_05" |
|---|---|
| Test Objective | Check that the IUT sends a Feedback for IncidentService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_FEEDBACK and CDM_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      Incident containing
        Identifier containing
```

```
          GeneratedIn indicating value any_value,
          UUID indicating value any_value,
          SickAnimalOnBoard indicating value "true", // Add test for the other features
          UrgencyType indicating value "Immediate",
          SeverityType indicating value "Moderate",
          Location  indicating value VESSEL_LOCATION,
          Agent containing
            identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value incidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Incident containing
              Identifier containing
                GeneratedIn indicating value any_value,
                UUID indicating value any_value,
                SickAnimalOnBoard indicating value "true", // Add test for the other features
                UrgencyType indicating value "Immediate",
                SeverityType indicating value "Moderate",
                Location  indicating value VESSEL_LOCATION,
                Agent containing
                  identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_06" |
|---|---|
| **Test Objective** | Check that the IUT sends a Feedback for IrregularMigrationIncident when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and<br>CDM_IRREGULAR_MIGRATION_INCIDENT_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      irregularMigrationIncident containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
```

```
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value irregularMigrationIncident,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            irregularMigrationIncident containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_07" |
|---|---|
| **Test Objective** | Check that the IUT sends a Feedback for LawInfringementIncidentService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.9 Incident Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and<br>CDM_LAW_INFRINGEMENT_INCIDENT_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      lawInfringementIncident containing
        identifier indicating value any_value,
        UrgencyType indicating value "Immediate",
          LawInfringementIncidentType indicating value "nonSpecified",
        Location indicating value VESSEL_LOCATION,
        Agent containing
          identifier indicating value AGENT_IDENTIFIER
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value lawInfringementIncidentService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            lawInfringementIncident containing
              identifier indicating value any_value,
              UrgencyType indicating value "Immediate",
              LawInfringementIncidentType indicating value "nonSpecified",
              Location indicating value VESSEL_LOCATION,
```

```
            Agent containing
              identifier indicating value AGENT_IDENTIFIER,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
            }
```

| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_08" |
|---|---|
| **Test Objective** | Check that the IUT sends a Feedback for MeteoOceanographicConditionService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_METEO_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      meteoOceanographicConditionService containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value meteoOceanographicConditionService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            meteoOceanographicConditionService containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_FEEDBACK_BV_09" |
|---|---|
| Test Objective | Check that the IUT sends a Feedback for OrganizationService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.10.2.2 MeteoOceanographicCondition Class<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_FEEDBACK and CDM_ORGANIZATION_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      organization containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value organizationService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            organization containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
            }
```

| TP Id | "TP_CDM_ADAPTOR_FEEDBACK_BV_10" |
|---|---|
| Test Objective | Check that the IUT sends a Feedback for RiskService when triggered |
| Reference | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.18 Risk Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| PICS Selection | IUT_CDM_NODE and CDM_FEEDBACK and CDM_RISK_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
            }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      risk containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
```

```
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value riskService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            risk containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

| **TP Id** | "TP_CDM_ADAPTOR_FEEDBACK_BV_11" |
|---|---|
| **Test Objective** | Check that the IUT sends a Feedback for CargoService when triggered |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism<br>ETSI GS CDM 005 [2], clause 7.1.6 Cargo Core Entity<br>ETSI GS CDM 005 [2], clause 7.1.19 UniqueIdentifier Core Entity |
| **PICS Selection** | IUT_CDM_NODE and CDM_FEEDBACK and CDM_CARGO_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
              }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT isTriggeredToSend a Feedback containing
      cargo containing
        IMONumber indicating value ANOMALY_IMO_NUMBER,
        Name indicating value ANOMALY_NAME,
        Location indicating value ANOMALY_LOCATION,
        ShipType indicating value ANOMALY_SHIP_TYPE
    from the TEST_SYSTEM entity
  }
  then {
    the IUT receives a vPOST containing
      uri indicating value CDM_FEEDBACK_REQUEST_URI,
      body containing
        vFeedback containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value true,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Feedback,
            ServiceType indicating value cargoService,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
```

```
          Payload containing
            cargo containing
              IMONumber indicating value ANOMALY_IMO_NUMBER,
              Name indicating value ANOMALY_NAME,
              Location indicating value ANOMALY_LOCATION,
              ShipType indicating value ANOMALY_SHIP_TYPE,
          Signature indicating value any_value,
          FeedbackType indicating value Info,
          Reason indicating value ShipTypeChange,
          RefMessageID indicating value REF_MESSAGE_ID
    to the NODE entity
  }
              }
```

## 6.5 Security

| **TP Id** | "TP_CDM_NODE_SEC_BV_01" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment with Success when it receives a sign message |
| **Reference** | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| **PICS Selection** | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          Signature containing
            SignedInfo containing
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
                SignatureMethod containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
                Reference_ indicating valueany_value,
                Transforms containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
                DigestMethod containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
                DigestValue indicating value any_value,
              SignatureValue indicating value any_value,
              KeyInfo containing
                X509Data containing
                  X509SubjectName indicating value SIGNING_CERT_SUBJECT,
                  X509Certificate indicating value SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value Success,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
        }
```

| TP Id | "TP_CDM_NODE_SEC_BO_01" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message with a corrupted signature |
| Reference | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| PICS Selection | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
      vPullRequest containing
        Signature containing
        SignedInfo containing
          CanonicalizationMethod containing
            Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
          SignatureMethod containing
            Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
          Reference_  indicating value  any_value,
          Transforms containing
            Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-signature",
          DigestMethod containing
            Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
          DigestValue indicating value any_value,
        SignatureValue indicating value corrupted_signature_value,
        KeyInfo containing
          X509Data containing
            X509SubjectName indicating value SIGNING_CERT_SUBJECT,
            X509Certificate indicating value SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value InvalidSignature,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
        }
```

| TP Id | "TP_CDM_NODE_SEC_BO_02" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message with a corrupted certificate |
| Reference | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| PICS Selection | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          Signature containing
            SignedInfo containing
```

```
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
              SignatureMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
              Reference_  indicating value  any_value,
              Transforms containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
              DigestMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
                DigestValue indicating value any_value,
              SignatureValue indicating value any_value,
              KeyInfo containing
                X509Data containing
                  X509SubjectName indicating value SIGNING_CERT_SUBJECT,
                  X509Certificate indicating value SIGNING_CERT
      from the ADAPTOR entity
    }
    then {
      the IUT sends a HTTP_RESPONSE containing
        status_code set to "200 OK"
        body containing
          vAcknwoledgment containing
            MessageID indicating value MESSAGE_ID,
            CorrelationID indicating value CORRELATION_ID,
            Priority indicating value PRIORITY,
            RequiresAck indicating value false,
            Sender containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull
            ,
            Recipient containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull
            ,
            AckCode indicating value InvalidSignature,
            Signature indicating value any_value



      to the ADAPTOR entity
    }
  }
```

| **TP Id** | "TP_CDM_NODE_SEC_BO_03" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message sign with corrupted digest |
| **Reference** | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| **PICS Selection** | IUT_CDM_NODE and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |
| with {<br>the IUT isInIdleState<br>} | |
| **Expected Behaviour** | |
| ensure that {<br>when {<br>the IUT receives a vPOST containing<br>uri indicating value CDM_PULL_REQUEST_URI,<br>body containing<br>vPullRequest containing<br>Signature containing<br>SignedInfo containing<br>CanonicalizationMethod containing<br>Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",<br>SignatureMethod containing<br>Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",<br>Reference_  indicating value  any_value,<br>Transforms containing<br>Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-signature",<br>DigestMethod containing<br>Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",<br>DigestValue indicating value corrupted_value,<br>SignatureValue indicating value any_value,<br>KeyInfo containing | |

```
              X509Data containing
                X509SubjectName indicating value SIGNING_CERT_SUBJECT,
                X509Certificate indicating value SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value InvalidSignature,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
        }
```

| **TP Id** | "TP_CDM_NODE_SEC_BO_04" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message sign with an unsupported signature algorithm |
| **Reference** | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| **PICS Selection** | IUT_CDM_NODE and CDM_VESSEL_SERVICE |
| **Initial Conditions** | |

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          Signature containing
            SignedInfo containing
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
              SignatureMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha384",
              Reference_  indicating value  any_value,
              Transforms containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
              DigestMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
              DigestValue indicating value any_value,
            SignatureValue indicating value any_value,
            KeyInfo containing
              X509Data containing
                X509SubjectName indicating value SIGNING_CERT_SUBJECT,
                X509Certificate indicating value SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
```

```
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value InvalidSignature,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
        }
```

| **TP Id** | "TP_CDM_NODE_SEC_BO_05" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message sign with an unsupported digest signature algorithm |
| **Reference** | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| **PICS Selection** | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          Signature containing
            SignedInfo containing
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
              SignatureMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
              Reference_  indicating value  any_value,
              Transforms containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
              DigestMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha384",
              DigestValue indicating value any_value,
            SignatureValue indicating value any_value,
            KeyInfo containing
              X509Data containing
                X509SubjectName indicating value SIGNING_CERT_SUBJECT,
                X509Certificate indicating value SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value InvalidSignature,
          Signature indicating value any_value
      to the ADAPTOR entity
  }
        }
```

| TP Id | "TP_CDM_NODE_SEC_BO_06" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message sign with an expired certificate (CDM_TS_CERTIFICATE_PAST) |
| Reference | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| PICS Selection | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          Signature containing
            SignedInfo containing
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
                SignatureMethod containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
                Reference_  indicating value  any_value,
                Transforms containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
                DigestMethod containing
                  Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
                DigestValue indicating value any_value,
              SignatureValue indicating value any_value,
              KeyInfo containing
                X509Data containing
                  X509SubjectName indicating value EXPIRED_SIGNING_CERT_SUBJECT,
                  X509Certificate indicating value EXPIRED_SIGNING_CERT
      from the ADAPTOR entity
    }
    then {
      the IUT sends a HTTP_RESPONSE containing
        status_code set to "200 OK"
        body containing
          vAcknwoledgment containing
            MessageID indicating value MESSAGE_ID,
            CorrelationID indicating value CORRELATION_ID,
            Priority indicating value PRIORITY,
            RequiresAck indicating value false,
            Sender containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull,
            Recipient containing
              ServiceID indicating value SERVICE_ID,
              ServiceOperation indicating value Pull,
            AckCode indicating value InvalidSignature,
            Signature indicating value any_value
      to the ADAPTOR entity
    }
        }
```

| TP Id | "TP_CDM_NODE_SEC_BO_07" |
|---|---|
| Test Objective | Check that the IUT responds with an HTTP 200 OK Acknowledgment with InvalidSignature when it receives a message sign with a certificate valid in the future (CDM_TS_CERTIFICATE_FUTURE) |
| Reference | ETSI GS CDM 004 [1], clause 8.1.10 Message Signature |
| PICS Selection | IUT_CDM_NODE and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
```

```
          Signature containing
            SignedInfo containing
              CanonicalizationMethod containing
                Algorithm indicating value "http://www.w3.org/2001/10/xml-exc-c14n#",
              SignatureMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#rsa-sha1",
              Reference_  indicating value  any_value,
              Transforms containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#enveloped-
signature",
              DigestMethod containing
                Algorithm indicating value "http://www.w3.org/2000/09/xmldsig#sha1",
              DigestValue indicating value any_value,
            SignatureValue indicating value any_value,
            KeyInfo containing
              X509Data containing
                X509SubjectName indicating value FUTURE_SIGNING_CERT_SUBJECT,
                X509Certificate indicating value FUTURE_SIGNING_CERT
    from the ADAPTOR entity
  }
  then {
    the IUT sends a HTTP_RESPONSE containing
      status_code set to "200 OK"
      body containing
        vAcknwoledgment containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          AckCode indicating value InvalidSignature,
          Signature indicating value any_value
    to the ADAPTOR entity
  }
        }
```

| **TP Id** | "TP_CDM_NODE_SEC_BO_08" |
|---|---|
| **Test Objective** | Check that the IUT responds with an HTTP 200 OK Acknowledgment response with InvalidSignature when it receives a message not signed |
| **Reference** | ETSI GS CDM 004 [1], clause 5.4.2 Pull<br>ETSI GS CDM 004 [1], clause 8.1.3 Pull (Request/Response)<br>ETSI GS CDM 004 [1], clause 8.1.4 Pull Request Querying Mechanism |
| **PICS Selection** | IUT_CDM_NODE and CDM_PULL and CDM_VESSEL_SERVICE |

**Initial Conditions**

```
with {
the IUT isInIdleState
        }
```

**Expected Behaviour**

```
ensure that {
  when {
    the IUT receives a vPOST containing
      uri indicating value CDM_PULL_REQUEST_URI,
      body containing
        vPullRequest containing
          MessageID indicating value MESSAGE_ID,
          CorrelationID indicating value CORRELATION_ID,
          CreationDateTime indicating value CURRENT_TIME,
          Priority indicating value PRIORITY,
          RequiresAck indicating value false,
          Sender containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Recipient containing
            ServiceID indicating value SERVICE_ID,
            ServiceOperation indicating value Pull,
          Payload containing
            Vessel containing
              IMONumber indicating value VESSEL_IMO_NUMBER,
              Name indicating value VESSEL_NAME,
              Location indicating value VESSEL_LOCATION,
```

```
            ShipType indicating value VESSEL_SHIP_TYPE,
        PullType indicating value Request,
        DiscoveryProfiles containing
          SeaBasin indicating value SEA_BASIN,
        not Signature
  from the ADAPTOR entity
}
then {
  the IUT sends a HTTP_RESPONSE containing
    status_code set to "200 OK"
    body containing
      vAcknwoledgment containing
        MessageID indicating value MESSAGE_ID,
        CorrelationID indicating value CORRELATION_ID,
        Priority indicating value PRIORITY,
        RequiresAck indicating value false,
        Sender containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        Recipient containing
          ServiceID indicating value SERVICE_ID,
          ServiceOperation indicating value Pull,
        AckCode indicating value InvalidSignature,
        Signature indicating value any_value
  to the ADAPTOR entity
}
      }
```

# History

| Document history | | |
|---|---|---|
| V1.1.1 | July 2024 | Publication |
| | | |
| | | |
| | | |
| | | |