DATEX II
EXCHANGE MECHANISM

European Commission DG TREN

Deliverable 4.1

Date	08/02/2005
Document Number	D2_4.1v1.01

Document Control:

Prepared by:			
DATEX II Team	**Date**	**Comment**	**Version Number**
Gerd Blake	23/04/2004	Created	0.1
Matthias Scholz	28/04/2004	Modified	0.2
Gerd Blake Matthias Scholz	25/05/2004	Revised	0.5
Kate McKemey	27/05/2004	Reviewed	0.5
Gerd Blake Matthias Scholz	27/05/2004	Reviewed	0.6
Matthias Scholz Gerd Blake	05/08/2004	completely restructured and revised	0.7
Matthias Scholz	10/09/2004	Addressing comments from RE	0.8
Matthias Scholz	12/11/2004	Addressing comments from RE	0.9
D2 Team /TC	18/11/2004	Review during the TC Workshop	0.11
Matthias Scholz	23/11/2004	Changes incorporated	0.13
Matthias Scholz	30/11/2004	Screenshots updated	1.00
Matthias Scholz	08/02/2005	Screenshots updates according to the datatype discussion	1.01

Reviewed by:			
DATEX II Team and DATEX TC	**Date**	**Comment**	**Version Number**
Robert Ellena	26/05/2004	WP4 TC leader comments on doc 20040525.doc	0.5
Robert Ellena	25/06/2004	20040625 WP41v06 comments 02.xls	0.6
Robert Ellena	27/08/2004	20040827 WP41v07 comments 05.xls	0.7
Robert Ellena	24/09/2004	20040921 WP41v08 comments 02.xls	0.8

Approved by:			
DATEX TC	**Date**	**Comment**	**Version Number**

TABLE OF CONTENTS

1. Introduction 2

2. DATEX 2 System Analysis 4
2.1. Basic System Overview 4
2.1.1. Supplier 4
2.1.2. Client 4
2.1.3. ManagementAdministrator 4
2.1.4. OperatingAdministrator 4
2.1.5. SubscriptionService 5
2.2. Sub Systems 5
2.2.1. PublisherSubSystem 5
2.2.2. ManagementSubSystem 5
2.2.3. DeliverySubSystem 5
2.3. Use Cases 5
2.3.1. Use Case “handleExchange” 6
2.3.2. Use Case “handleData” 6
2.3.3. Use Case “handleSubscription” 6
2.3.4. Use Case “managePartner” 7
2.3.5. Use Case “manageFilter” 7
2.3.6. Use Case “manageCatalogue” 7

3. Exchange Meta Data 9
3.1. Class Diagram Of Exchange Meta Data 9

4. Exchange Mechanism Overview 12
4.1. Operating Modes 12
4.2. Update Methods 12
4.3. Exchange Services 12
4.4. Session Management 13

5. Exchange Specification 15
5.1. Preconditions 15
5.2. Operating Mode 0 – Subscription Management Mechanism 16
5.2.1. Session Service 18
5.2.2. Subscription Service 19
5.2.2.1 Create Subscription 19
5.2.2.2 Edit Subscription 21
5.2.2.3 Delete Subscription 25
5.2.3. Catalogue Publication Service 26
5.2.4. Filter Publication Service 28
5.2.4.1. Filter Operation Acknowledgement 28
5.2.4.2. Create Customer Defined Filters 30
5.2.4.3. Delete Customer Defined Filters 31
5.3. Operating Mode 1 – Supplier Push On Occurrence 32
5.4. Operating Mode 2 – Supplier Push Periodic 34
5.5. Operating Mode 3 – Client Pull 35

6. Annex 38
6.1. Interchange Agreement 38
6.1.1. General 38
6.1.2. Access 38
6.1.3. Data exchange 38
6.1.4. Rights and obligations 38
6.2. Table Of Figures 39

INTRODUCTION

1. Introduction

This document specifies an exchange mechanism for DATEX 2 Messages. Static attributes are included to enable a DATEX 2 message exchange as well as a dynamic view of the exchange mechanism. The document also provides a general system overview.

There are three types of exchange envisaged. These comprise a “fire & forget”, filter type exchange and a catalogue type exchange. In a “fire & forget” type exchange the Client would receive a “snapshot” of the information the Supplier has agreed to provide. A filter type exchange would be one where one or more filters are provided to the Supplier for implementation on the Client’s behalf. A catalogue type exchange would be one where the Supplier initially provides one or more catalogues containing lists of available information. The Client can then build one or more subscriptions comprising items from those catalogues. This is then returned to the Supplier, who then implements the Client’s subscription requests.

DATEX 2 SYSTEM OVERVIEW

D2 evolving DATEX

2. DATEX 2 System Analysis

2.1. Basic System Overview

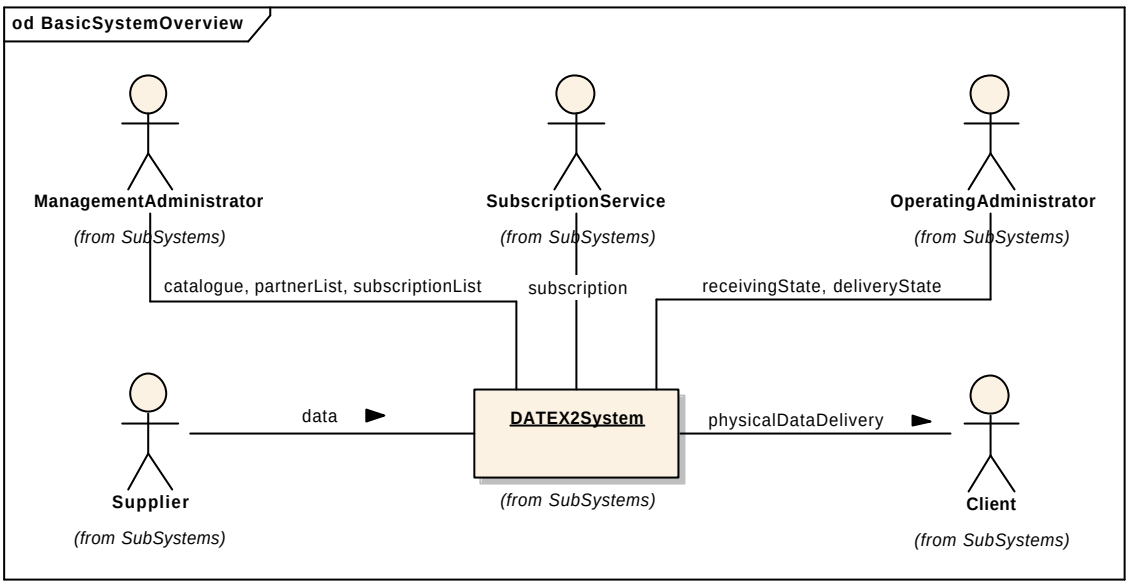

Figure 1 – Analysis Diagram "BasicSystemOverview"

The above figure provides a basic system overview, including actors able to interact with the system. Five Actors dealing with the DATEX 2 System are identified:

- Supplier
- Client
- ManagementAdministrator
- OperatingAdministrator
- SubscriptionService

> A DATEX 2 System can interact with several DATEX 2 Clients.

All actors are described in detail below.

2.1.1. Supplier

- provides data as input to the DATEX 2 System
- is either a system supporting exchange of DATEX 2 Messages or a human operator who enters data manually into the DATEX 2 System
- is mandatory

2.1.2. Client

- receives traffic and travel related data from the DATEX 2 System
- can be either another DATEX 2 System or a simple Client
- is mandatory for the exchange process

2.1.3. ManagementAdministrator

- can be a human or a system
- manages the list of authorized Users
- creates and maintains a catalogue of the available data
- creates and maintains the available filters
- monitors and maintains the subscription list
- is mandatory

2.1.4. OperatingAdministrator

- can be a human or a system
- monitors the data input from the Supplier
- monitors the physical data delivery status
- surveys the quality of the input and delivery
- is optional

2.1.5. SubscriptionService

- can be an integrated component or an external service
- can be done either by the Client (online) or by the ManagementAdministrator (off-line)
- enters subscriptions
- is mandatory

2.2. Sub Systems

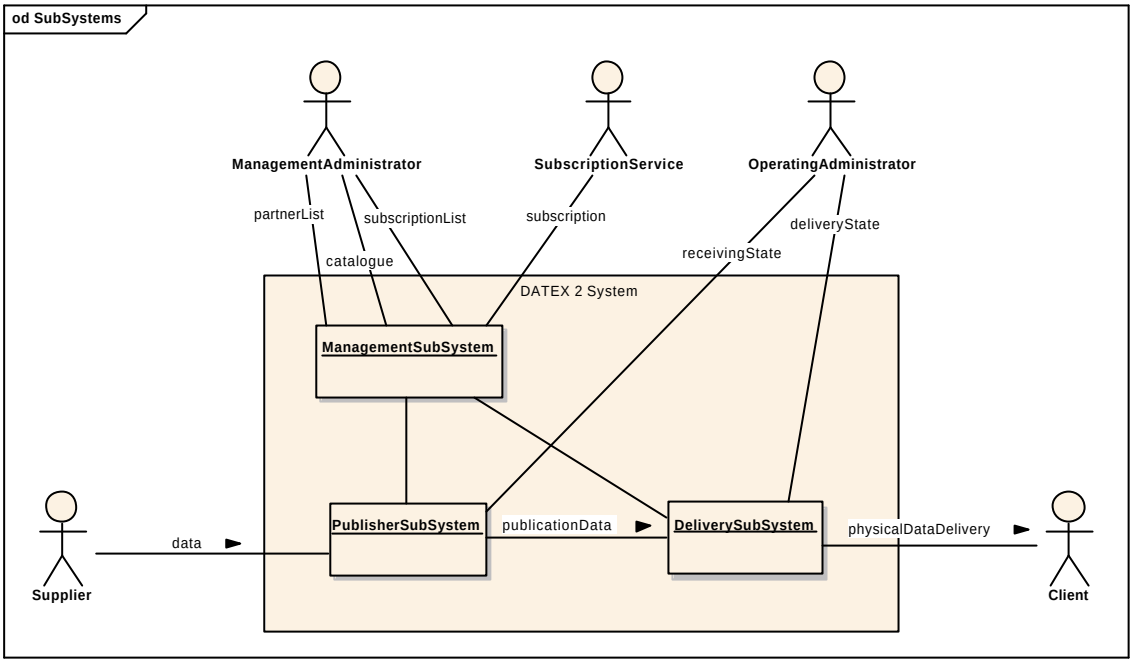

Figure 2 – Analysis Diagram "SubSystem"

The DATEX2 System consists of PublisherSubSystem, DeliverySubSystem and ManagementSubSystem.

2.2.1. PublisherSubSystem

- makes data available
- creates the payload package
- interacts with the DeliverySubSystem
- mandatory

2.2.2. ManagementSubSystem

- manages the client, catalogue and subscription lists
- mandatory (at least for subscriptions)

2.2.3. DeliverySubSystem

- adds exchange specific information
- performs the physical delivery
- interacts with the Client
- mandatory

2.3. Use Cases

The use case diagram below specifies the general capabilities of the DATEX 2 System. Special attention is drawn to use cases

- providing functionalities to interact with the client
- which influence use cases necessary to interact with the client

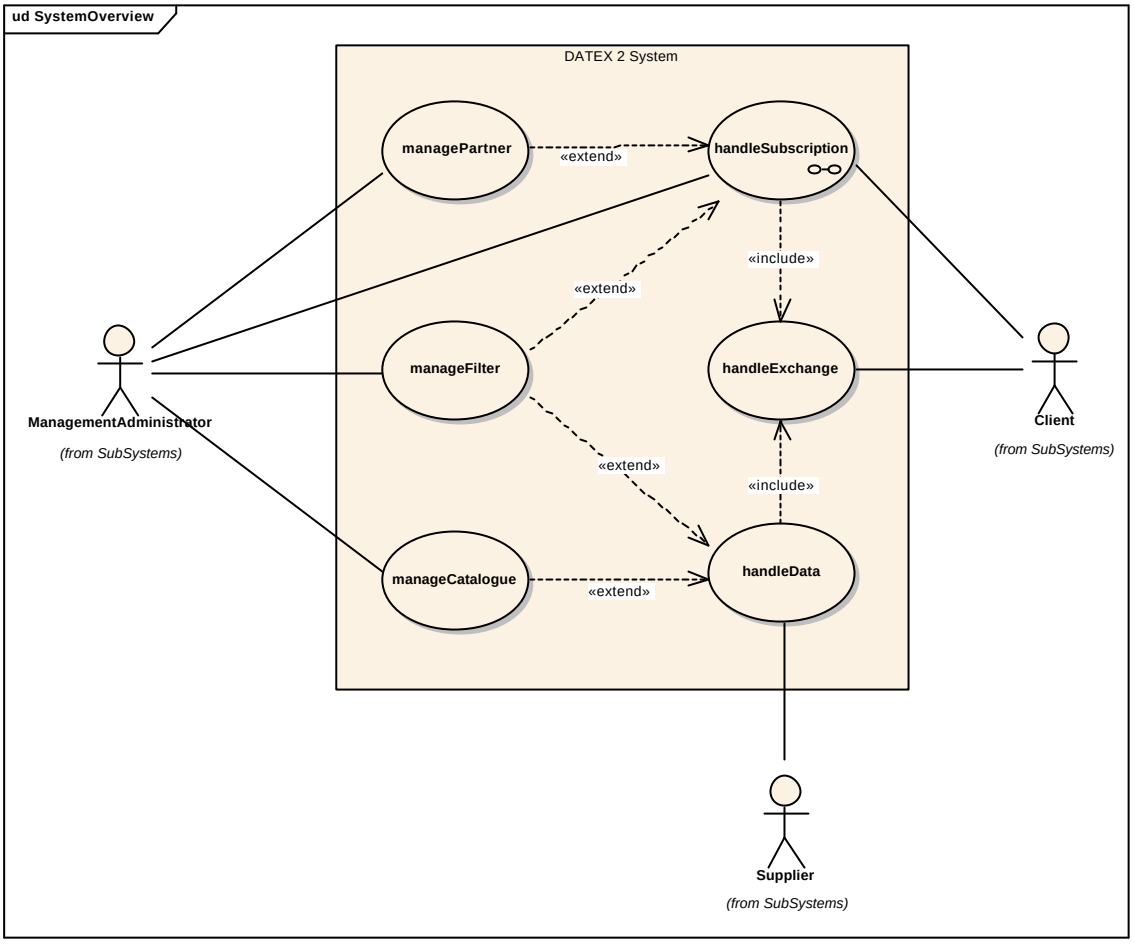

Figure 3 – Use Case Diagram "SystemOverview"

As an example “extend” here means, that manageFilter uses handleSubscription to exchange filters when required.

2.3.1. Use Case “handleExchange”

This is the key use case for interacting with the “Client”.
“handleExchange” provides a complete set of functionalities for exchanging data. It includes “handleSubscription” and “handleData”.
“handleSubscription” provides the functionality dealing with the subscription management mechanism. This is a necessary pre-condition for a data exchange.
“handleData” provides the functionality to manage the payload, the data to be exchanged.
“managePartner”, “manageFilter” and “manageCatalogue” extend these use cases to enrich the functionality.

Pre-condition: A valid subscription and suitable payload is available.
Post-condition: Data is successfully delivered to the client.

2.3.2. Use Case “handleData”

“handleData” provides the functionality to manage data to be exchanged. It interacts with actor ”Supplier”. Catalogues and filters are part of the payload. Therefore “handleData” is extended by the use cases “manageCatalogue” and “manageFilter”.

Pre-condition: The supplier provides data to the system.
Post-condition: Data are available.

2.3.3. Use Case “handleSubscription”

“handleSubscription” provides the functionality for the subscription management mechanism. A subscription is a necessary pre-condition for a data exchange. The following use case diagram details the view on “handle”Subscription” and is included because of the importance of a subscription.

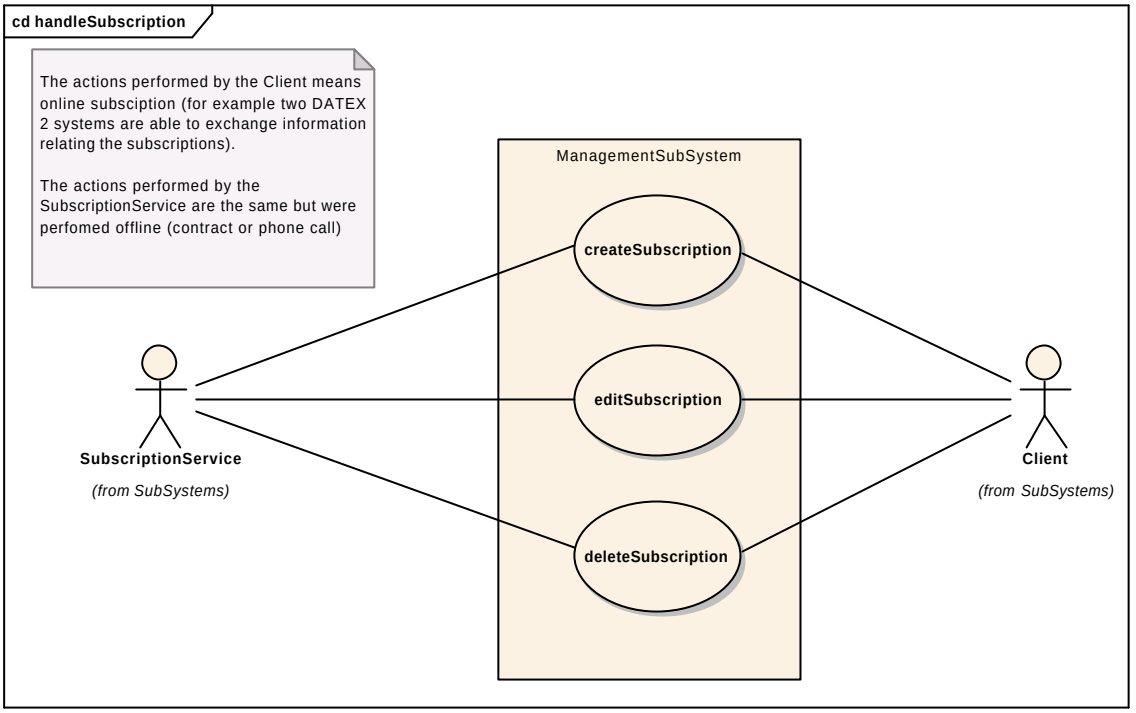

Figure 4 – Use Case Diagram "HandleSubscription"

The above figure illustrates how “handleSubscription” provides functionality to create, edit and delete a subscription.

Pre-condition: Either “SubscriptionService” or “Client” is able to interact with the system in order to manage subscriptions.
Post-condition: Subscriptions are successfully handled.

2.3.4. Use Case “managePartner”
“managePartner” provides the functionality to manage both clients and suppliers. It contains information about usernames and passwords.

2.3.5. Use Case “manageFilter”
“manageFilter” provides the functionality to manage filters. Creating, editing or deleting filters can either be done by the “ManagementAdministrator” or directly by a client using the exchange mechanism. Filters themselves are handled as payload.

2.3.6. Use Case “manageCatalogue”
“manageCatalogue” provides the functionality to manage Catalogues. The “ManagementAdministrator” is able to create, edit and delete either the entire catalogue or separate catalogue items.

EXCHANGE META DATA

3. Exchange Meta Data

This section deals with the modelling of the PIM Exchange Mechanism. The model itself is available as an Enterprise Architect Project file (.EAP). Please refer to this file for a comprehensive view of the model.

3.1. Class Diagram Of Exchange Meta Data

To fully specify delivery of traffic and travel related information a set of exchange metadata is needed. The following UML class diagram illustrates this.

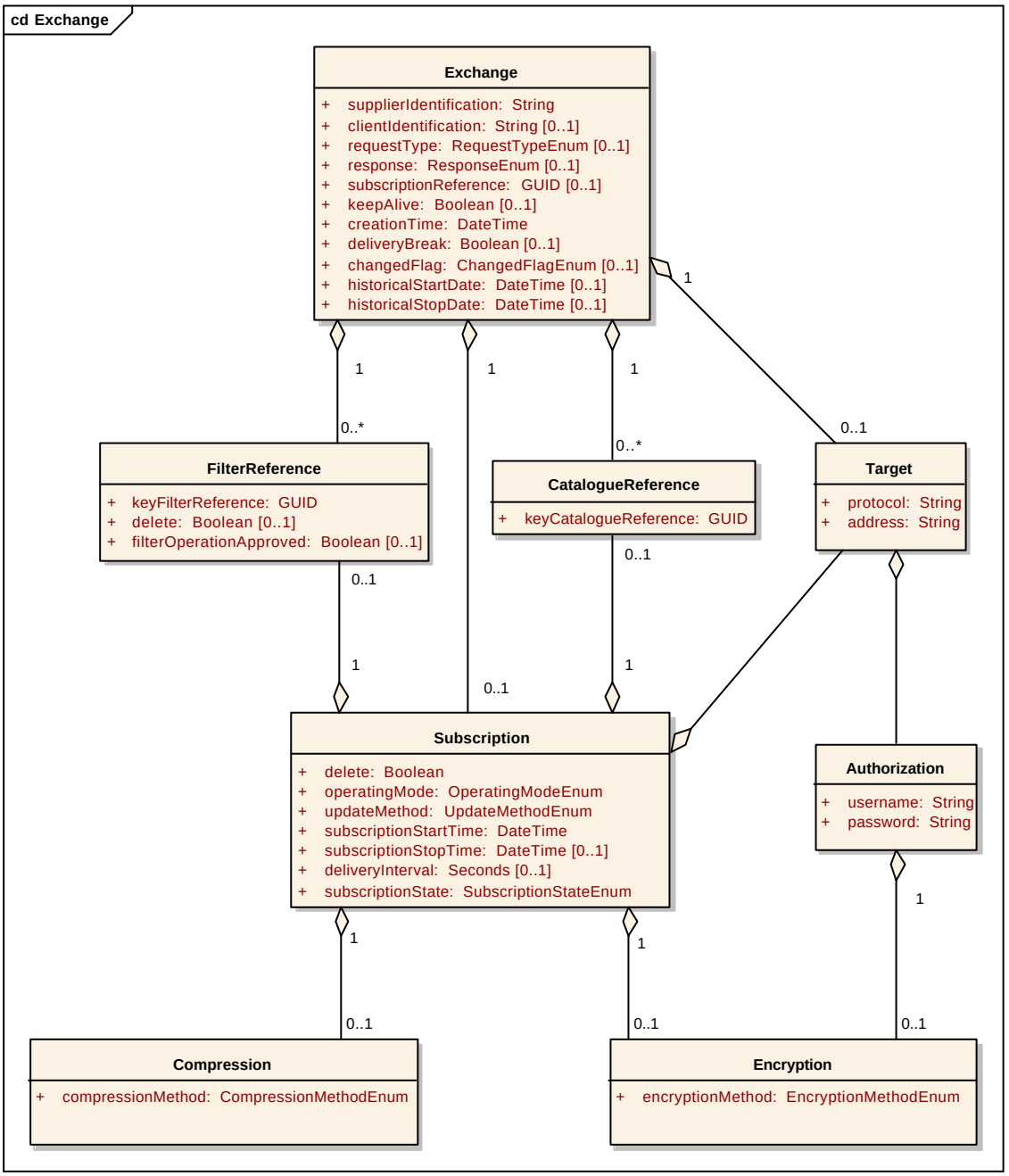

Figure 5 – Class Diagram "Exchange"

The class **Exchange** is part of every delivery but may use different attributes. Many of the elements marked as here as optional may become mandatory elements in certain cases such as catalogue or filter requests. The usage of this metadata is subsequently described in detail.

In addition all necessary enumeration classes are shown in the diagram below.

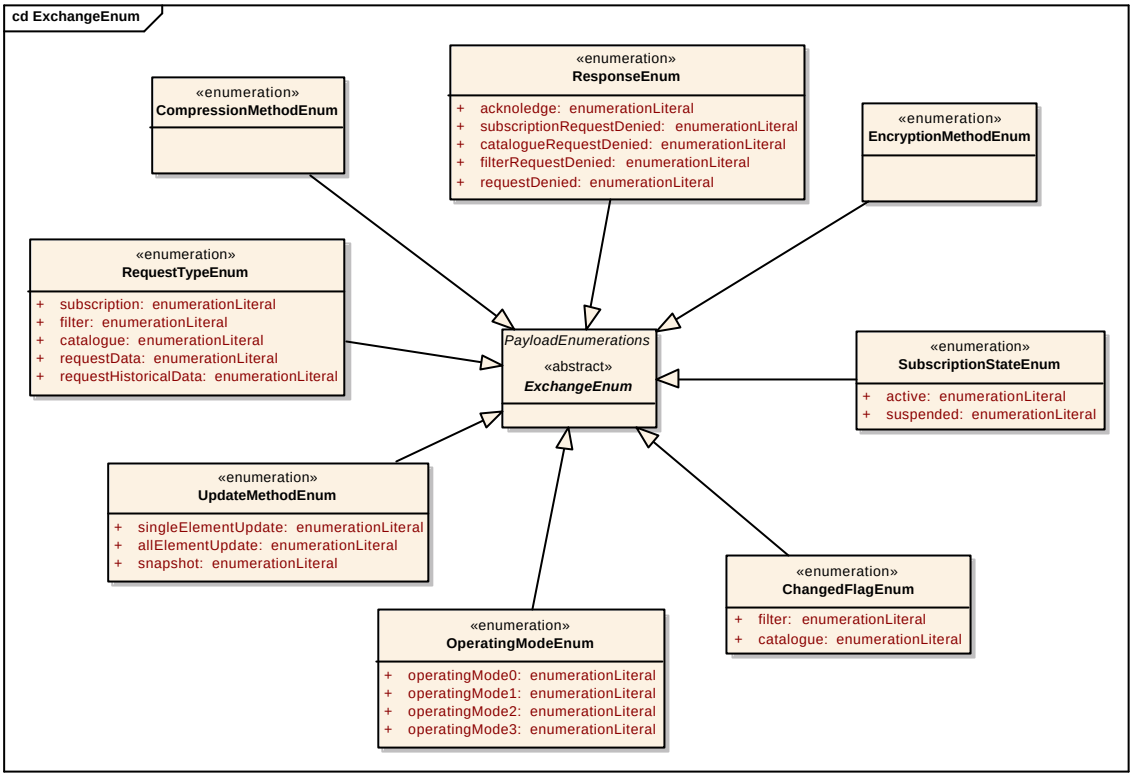

Figure 6 – Class Diagram "ExchangeEnum"

EXCHANGE MECHANISM OVERVIEW

4. Exchange Mechanism Overview

This chapter provides a high level overview of the DATEX 2 Exchange Mechanism.

4.1. Operating Modes

An Operating Mode is a set of rules and conditions that stipulate the physical transmission of data between supplier and client(s)

- Operating Mode 0 – Subscription Management Mechanism
  - a specialized Operating Mode to handle subscriptions
- Operating Mode 1 – Publisher Push on occurrence
  - data delivery initiated by the publisher every time data is changed
- Operating Mode 2 – Publisher Push periodic
  - data delivery initiated by the publisher on a cyclic time basis
- Operating Mode 3 – Client Pull
  - data delivery initiated by the Client, where data is returned as a response.

Each Operating Mode can be both on- and offline.

4.2. Update Methods

Differing amounts of data can be delivered within a Publication. Dependent on the type of Publication the following Update Methods are possible:

- singleElementUpdate
  - If an atomic part of the data has been changed this atomic part, and only this atomic part, will be exchanged.

- allElementUpdate
  - If an atomic part has been changed the data complex associated with the atomic part will be exchanged.

- Snapshot
  - A snapshot contains all information that is available for a subscription..

4.3. Exchange Services

A class diagram containing only abstract classes has been modelled to illustrate the physical exchange of data. With the help of these abstract classes the notion of Services is introduced. From a base class three different services are derived as shown in the picture below.

> The modelling shown in the diagram below is the only way to illustrate the common concept for both simple and sophisticated mechanisms.

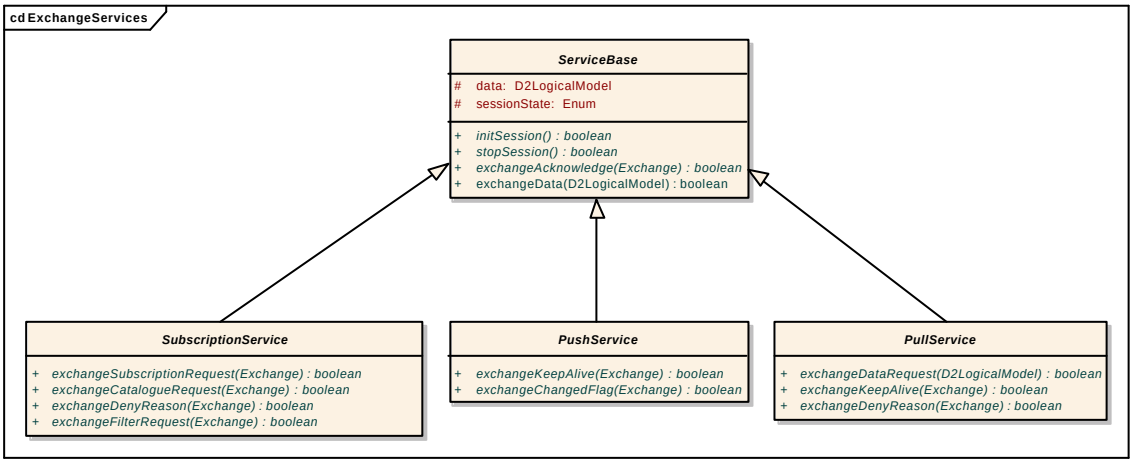

Figure 7 – Class Diagram “ExchangeServices”

This diagram is valid for both Client and Supplier systems. For realization it may be necessary to build particular specializations for client and the supplier system.

The model represents the abstract concept of an exchange; its implementation will depend on the particular technical realisation. This may include defining supplementary methods within the given classes.

All operations which are shown in the class diagram above can be performed either on- or offline. Particularly for Subscription the following levels of automation may be possible:

- full offline process
- semi-automated process
- full online based subscription

4.4. Session Management

Logical sessions are a vital pre-condition for session management. Each service has the ability to start and to stop a session. Therefore, and with the help of a keep-alive mechanism, observation of the session state is possible.

At system start up or after link failure, there could be a need for synchronizing data. The recovery mechanism depends on the chosen Operating Mode.

For Operating Mode 1 and 2 this should be agreed upon by client and supplier. One possible solution could be to use a combination of subscriptions for example first using Operating Mode 3 – Client Pull, in order to get a full set of data followed by a subscription using the previously agreed Operating Mode. In such a case network traffic could be considerable.

For Operating Mode 3 this issue does not arise.

EXCHANGE SPECIFICATION

D2 evolving DATEX

5. Exchange Specification

5.1. Preconditions

A subscription is a necessary pre-condition for the exchange of data. Without a subscription delivery of data is not possible.

To enable a Client to make a subscription, either online or offline, contact information is required. In either case, offline through a contact person or online automated, the following information is necessary:

- mandatory: address of the DATEX 2 System to which the clients wants to connect
- optional: username
- optional: password
- optional: encryption method for username and password during the transmission

In the case that username and password are required, the DATEX 2 System will need to hold this information in a partner list for authentication.

5.2. Operating Mode 0 – Subscription Management Mechanism

The figure below shows the Operating Mode 0 – Subscription Management Mechanism. It is used to create, edit and delete subscriptions.

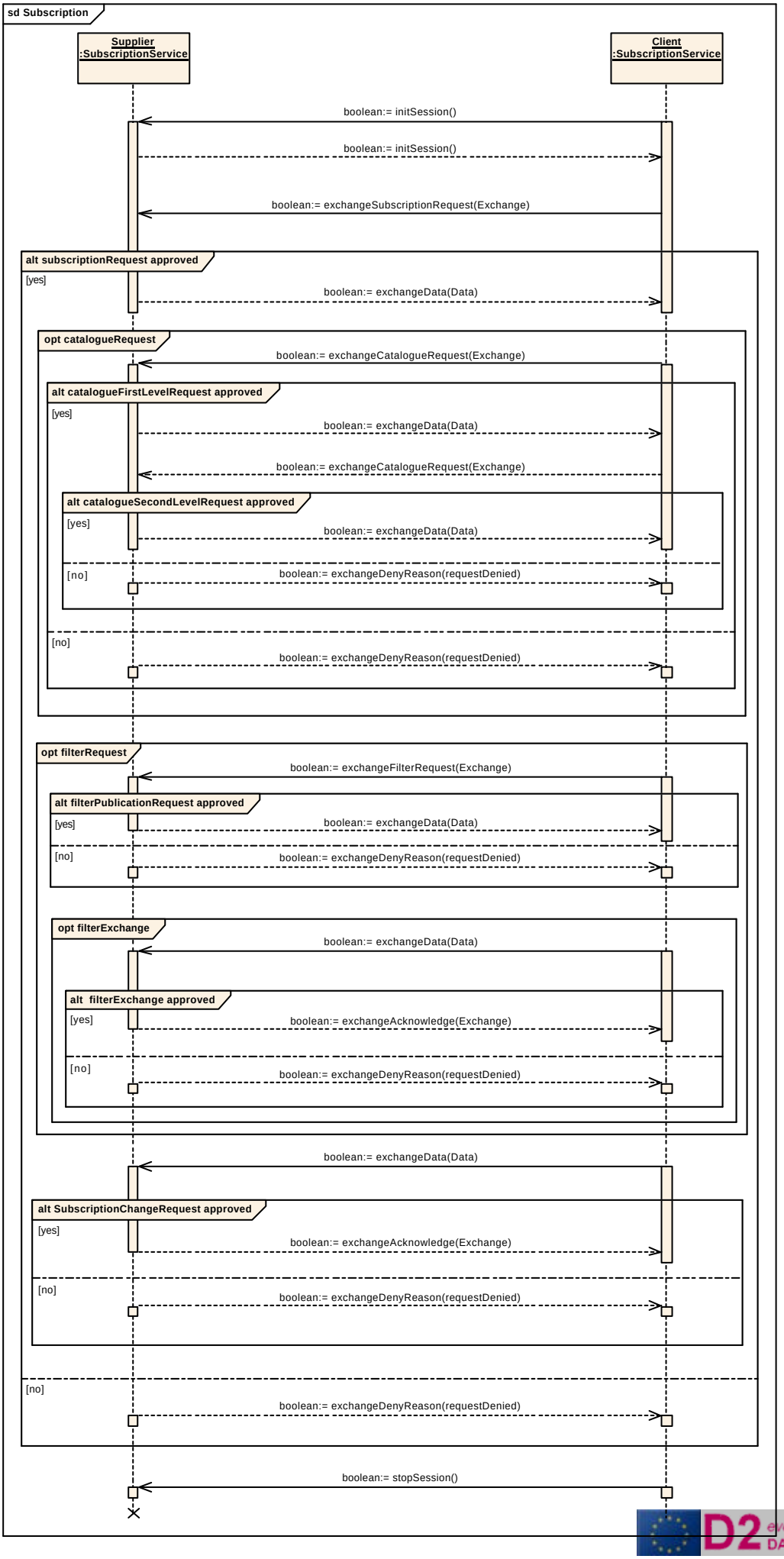
sd Subscription
Supplier
:SubscriptionService
Client
:SubscriptionService
boolean:= initSession()
boolean:= initSession()
boolean:= exchangeSubscriptionRequest(Exchange)
alt subscriptionRequest approved
[yes]
boolean:= exchangeData(Data)
opt catalogueRequest
boolean:= exchangeCatalogueRequest(Exchange)
alt catalogueFirstLevelRequest approved
[yes]
boolean:= exchangeData(Data)
boolean:= exchangeCatalogueRequest(Exchange)
alt catalogueSecondLevelRequest approved
[yes]
boolean:= exchangeData(Data)
[no]
boolean:= exchangeDenyReason(requestDenied)
[no]
boolean:= exchangeDenyReason(requestDenied)
opt filterRequest
boolean:= exchangeFilterRequest(Exchange)
alt filterPublicationRequest approved
[yes]
boolean:= exchangeData(Data)
[no]
boolean:= exchangeDenyReason(requestDenied)
opt filterExchange
boolean:= exchangeData(Data)
alt filterExchange approved
[yes]
boolean:= exchangeAcknowledge(Exchange)
[no]
boolean:= exchangeDenyReason(requestDenied)
boolean:= exchangeData(Data)
alt SubscriptionChangeRequest approved
[yes]
boolean:= exchangeAcknowledge(Exchange)
[no]
boolean:= exchangeDenyReason(requestDenied)
[no]
boolean:= exchangeDenyReason(requestDenied)
boolean:= stopSession()

Figure 8 – Sequence Diagram "Subscription"

5.2.1. Session Service

In the diagram below the mandatory phase of initializing a session is shown. Using sessions makes active link management possible.

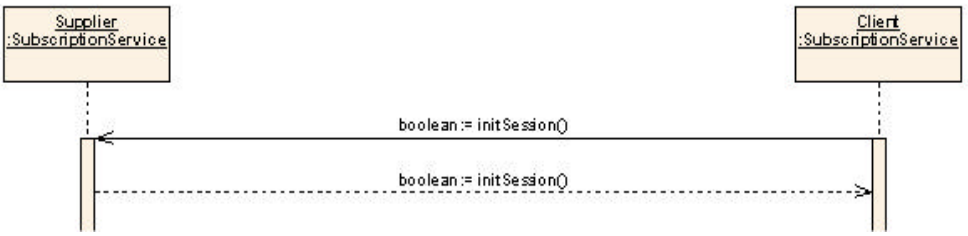

Figure 9 – Sequence Diagram Part “initSession”

A session is initialized by the client through opening a physical connection to the Supplier. The Supplier acknowledges the connection by using the “initSession” operation. This establishes a logical session.

Both the Client and the Supplier are able to stop the Session as shown in the diagram below at any time.

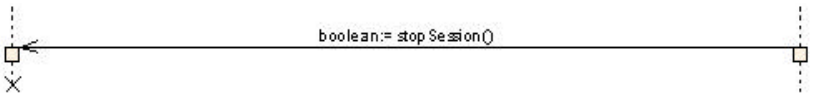

Figure 10 – Sequence Diagram Part “stopSession”

5.2.2. Subscription Service

A Client subscription relates to a single catalogue reference. It is therefore expected that a Client may use one or more subscriptions.

5.2.2.1 Create Subscription

After establishing a session, Client and Supplier have to exchange subscription information.

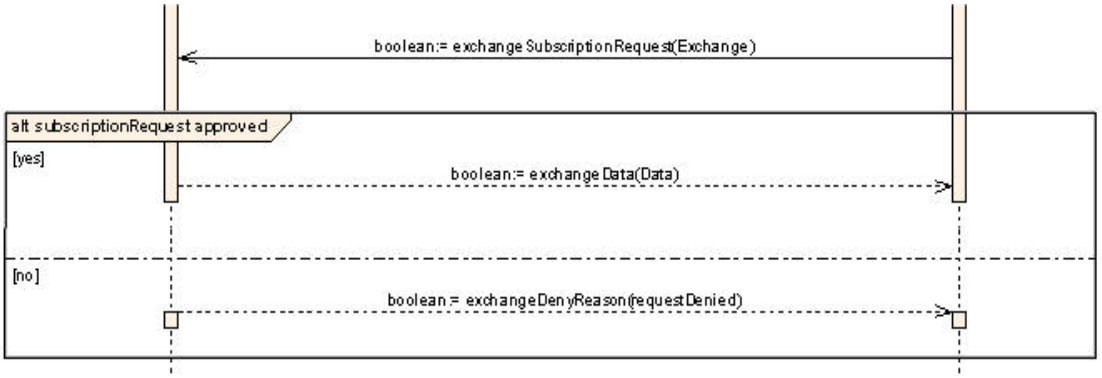

Figure 11 – Sequence Diagram Part “subscriptionRequest”

The Client sends an exchangeSubscriptionRequest message to the Supplier which can either be approved by sending a subscription identification or not approved transmitting a DeniedReason.

The **exchangeSubscriptionRequest(Exchange)** has mandatory elements and may contain optional elements as outlined in the object form.

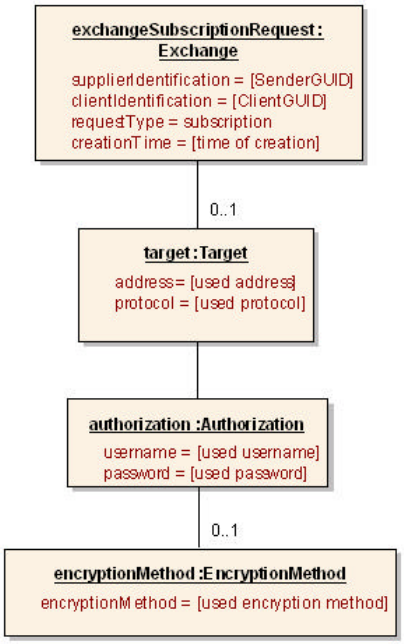

Figure 12 – Create Subscription

The following XML code is provided as an example for creating a subscription:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <requestType>subscription</requestType>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <Target>
                  <protocol>HTTP</protocol>
                  <address>http://192.168.36.15</address>
                  <Authorization>
                        <username>guest</username>
                        <password>guest</password>
                  </Authorization>
            </Target>
      </Exchange>
```

```
</D2LogicalModel>
```

The **exchangeDenyReason** message has the following mandatory elements as outlined in the object form.

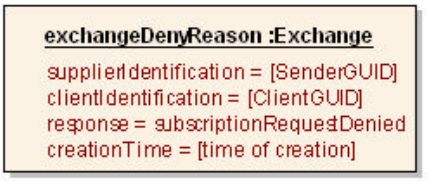

Figure 13 – Exchange Deny Reason Reference

The following XML code is provided as an example for a subscription denial:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>subscriptionRequestDenied</response>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The **exchangeData** message has the following mandatory elements as outlined in the object form.

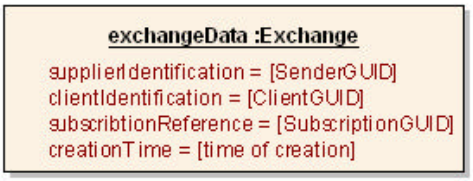

Figure 14 – Exchange Subscription Reference

The following XML code is provided as an example for receiving a subscription reference:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The sample above only describes how to create an empty subscription. In order to fill this empty subscription it has to be completed by using the “editSubscription” use case described below.

5.2.2.2 Edit Subscription

A Client is able to change the subscription once the reference GUID has been received as shown in the diagram.

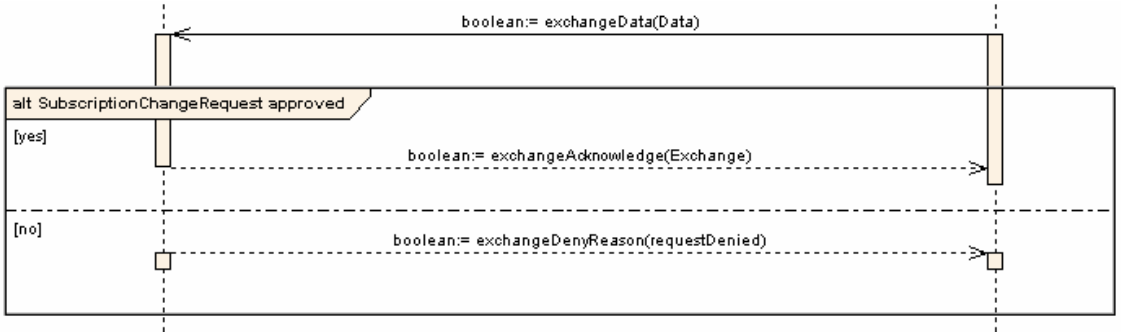

Figure 15 – Edit Subscription Mechanism

The **exchangeData** message amends a subscription by sending all subscription related information to the Supplier. Information about Catalogues and Filters has to be exchanged in advance. These mechanisms are described in subchapters 5.2.3 and 5.2.4.

A service commences immediately after the subscriptionStartTime and runs until the optional subscriptionStopDateTime.

The DATEX 2 System either approves a subscription by sending the exchangeAcknowledge message or denies the subscription by using the exchangeDenyReason message.

The mandatory elements and the range of values are specified by the selected Operating Mode.

5.2.2.2.1. Edit Subscription For Operating Mode 1

The **exchangeData** message for Operating Mode 1 has the following mandatory elements as outlined in the object form.

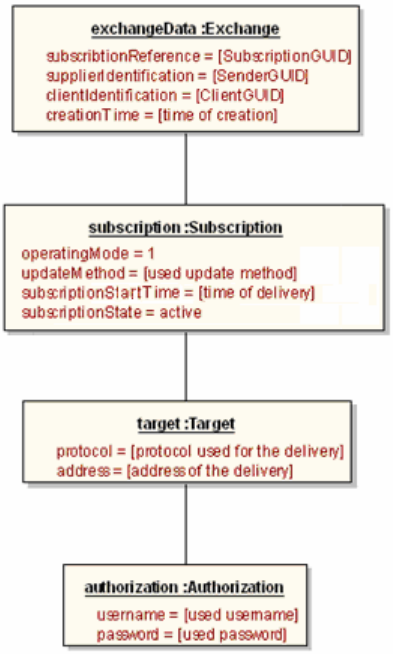

Figure 16 – Exchange Data Reference

In addition the following optional elements can be used

- Exchange
  - Subscription
    - subscriptionStopTime
    - CatalogueReference
    - FilterReference
    - CompressionMethod
    - EncryptionMethod
    - Target
      - Authorization

- EncryptionMethod

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <Subscription>
                  <operatingMode>1</operatingMode>
                  <updateMethod>singleElementUpdate</updateMethod>
                  <subscriptionStartTime>
                        2004-08-15T08:00:00
                  </subscriptionStartTime>
                  <subscriptionState>active</subscriptionState>
                  <FilterReference>
                        <keyFilterReference>
                              selectedFilterGUID
                        </keyFilterReference>
                  </FilterReference>
                  <Target>
                        <protocol>HTTP</protocol>
                        <address>http://192.168.40.15</address>
                        <Authorization>
                              <username>guest</username>
                              <password>guest</password>
                        </Authorization>
                  </Target>
            </Subscription>
      </Exchange>
</D2LogicalModel>
```

5.2.2.2.2. Edit Subscription For Operating Mode 2

The exchangeData message for the Operating Mode 2 has the following mandatory elements

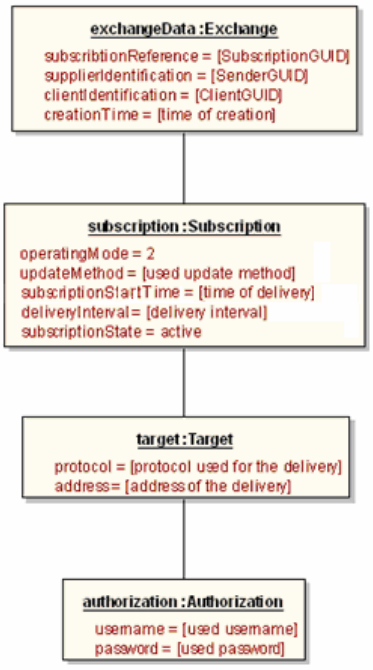

Figure 17 – Exchange Data Reference

In addition the following optional elements can be used

- Exchange
  - Subscription
    - subscriptionStopTime
    - CatalogueReference
    - FilterReference
    - CompressionMethod
    - EncryptionMethod
    - Target
      - Authorization
        - EncryptionMethod

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <Subscription>
                  <operatingMode>2</operatingMode>
                  <updateMethod>snapshot</updateMethod>
                  <subscriptionStartTime>
                        2004-08-15T08:00:00
                  </subscriptionStartTime>
                  <subscriptionStopTime>2004-12-31T20:00:00</subscriptionStopTime>
                  <deliveryInterval>
                        600
                  </deliveryInterval>
                  <subscriptionState>active</subscriptionState>
                  <CatalogueReference>
                        <keyCatalogueReference>
                              selectedCatalogueGUID
                        </keyCatalogueReference>
                  </CatalogueReference>
                  <FilterReference>
                        <filterReference>
                              selectedFilterGUID
                        </filterReference>
                  </FilterReference>
                  <Target>
                        <protocol>HTTP</protocol>
                        <address>http://192.168.36.15</address>
                        <Authorization>
                              <username>guest</username>
                              <passoword>guest</password>
                        </Authorization>
                  </Target>
            </Subscription>
      </Exchange>
</D2LogicalModel>
```

5.2.2.2.3. Edit Subscription For Operating Mode 3

The exchangeData message for the Operating Mode 3 has the following mandatory elements

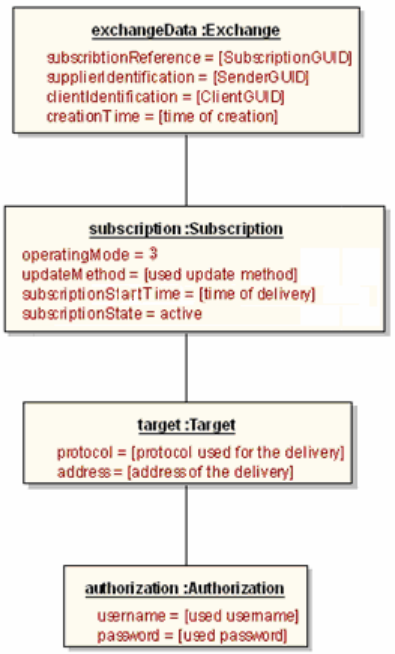

Figure 18 – Exchange Data Reference

In addition the following optional elements may be used

- Exchange
  - Subscription
    - subscriptionStopTime

- CatalogueReference
- FilterReference
- CompressionMethod
- EncryptionMethod
- Target
  - Authorization
    - EncryptionMethod

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <Subscription>
                  <operatingMode>3</operatingMode>
                  <updateMethod>singleElementUpdate</updateMethod>
                  <SubscriptionStartTime>2004-08-15T08:00:00</subscriptionStartTime>
                  <subscriptionState>active</subscriptionState>
                  <CatalogueReference>
                        <keyCatalogueReference>
                              selectedCatalogueGUID
                        </keyCatalogueReference>
                  </CatalogueReference>
                  <Target>
                        <protocol>HTTP</protocol>
                        <address>http://192.168.37.196</address>
                        <Authorization>
                              <username>guest</username>
                              <password>guest</password>
                        </Authorization>
                  </Target>
            </Subscription>
      </Exchange>
</D2LogicalModel>
```

The exchangeAcknowledge message has the following mandatory elements

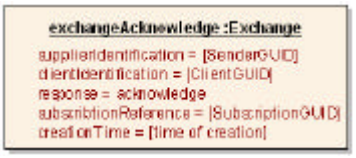

Figure 19 – Exchange Acknowledge Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>acknowledge</response>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The exchangeDenyReason message has to have the following mandatory elements.

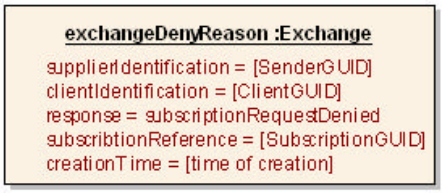

Figure 20 – Exchange Deny Reason Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>subscriptionRequestDenied</response>
```

```
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

5.2.2.3 Delete Subscription

Deleting a subscription is done by using the Edit Subscription Mechanism with the element “delete” set to “true”. To delete a subscription, send a message containing all the values of the particular subscription, which is to be deleted from the DATEX 2 System, together with the delete element.

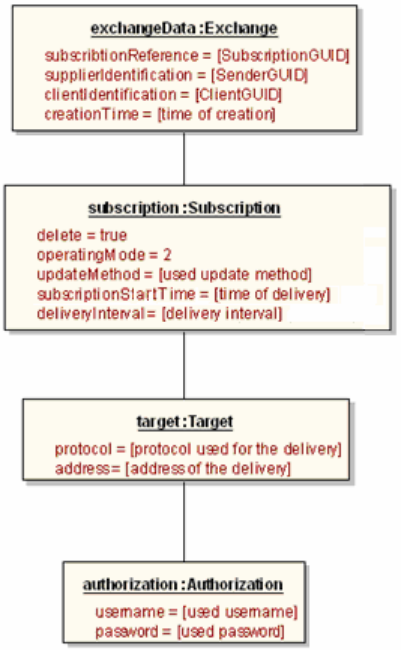

Figure 21 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <Subscription>
                  <delete>true</delete>
                  <operatingMode>2</operatingMode>
                  <updateMethod>snapshot</updateMethod>
                  <startTime>2004-08-15T08:00:00</startTime>
                  <subscriptionStopTime>2004-12-31T20:00:00</subscriptionStopTime>
                  <deliveryInterval>
                        600
                  </deliveryInterval>
                  <subscriptionState>active</subscriptionState>
                  <CatalogueReference>
                        <keyCatalogueReference>
                              selectedCatalogueGUID
                        </keyCatalogueReference>
                  </CatalogueReference>
                  <FilterReference>
                        <keyFilterReference>
                              selectedFilterGUID
                        </keyFilterReference>
                  </FilterReference>
                  <Target>
                        <protocol>HTML</protocol>
                        <address>http://192.168.36.15/DATEX2</address>
                        <Authorization>
                              <username>guest</username>
                              <passoword>guest</password>
                        </Authorization>
                  </Target>
            </Subscription>
      </Exchange>
</D2LogicalModel>
```

If the deletion was successful, the DATEX 2 System will send the acknowledge message otherwise the DATEX 2 System returns the requestDenied message, including the deny reason as described in the preceding chapter.

5.2.3. Catalogue Publication Service

When a catalogue service is used Clients must ensure they hold the same version of catalogues held by the Supplier. To ensure this, the Client must first request what versions of the catalogues are used by the Supplier. With this information a Client is able to assess whether it is necessary to request an update of catalogues, which are defined in the CataloguePublication Package. By sending the relevant catalogue reference, the Client can then request details of individual catalogue entries.

Figure 22 – Catalogue Publication Service

Where use is made of the **optional** Catalogue Publication Service, the Client sends an exchangeCatalogueRequest message, which can be either be approved by sending the first level of the CataloguePublication within the exchangeData message, or denied by sending the exchangeDenyReason message.

If one of the catalogues has been changed, the element changedFlag shall be used. In this case, only the Exchange class with its mandatory elements and the element changedFlag set to "catalogue" is sent. After receiving the information, the client can request the updated catalogue.

The exchangeCatalogueRequest message has to have the following mandatory elements.

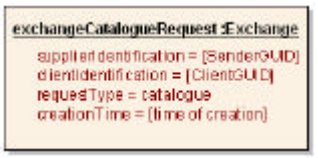

Figure 23 – Exchange Catalogue Request Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <requestType>catalogue</requestType>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

exchangeData messages have the following mandatory elements

- Payload
  - CataloguePublication with all first level catalogue entries
- the elements shown in the object form:

Figure 24 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            <CataloguePublication>
                  …
            </CataloguePublication>
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The exchangeDenyReason message has the following mandatory elements.

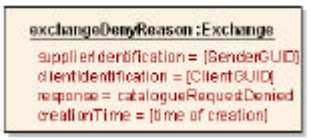

Figure 25 – Exchange Deny Reason Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>catalogueRequestDenied</response>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

After receiving the first level of the cataloguePublication the Client can select the catalogue required by sending a exchangeCatalogueRequestmessage with the following elements:

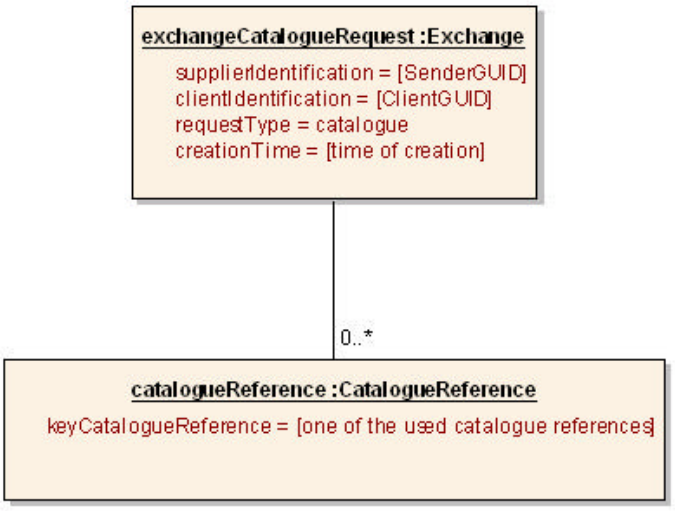

Figure 26 – Exchange Catalogue Request Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <requestType>catalogue</requestType>
            <creationTime>2004-08-14T11:06:28</creationTime>
             <CatalogueReference>
                   <keyCatalogueReference>wantedCatalogueGUID</keyCatalogueReference>
             </CatalogueReference>
```

```
            <CatalogueReference>
                    <keyCatalogueReference>wantedCatalogueGUID</keyCatalogueReference>
             </CatalogueReference>
            <catalogueReference>wantedCatalogueGUID</catalogueReference>
            <catalogueReference>wantedCatalogueGUID</catalogueReference>
      </Exchange>
</D2LogicalModel>
```

This example shows the request for two catalogues in details.

The DATEX 2 System can approve the request by sending the requested catalogues using the exchangeData message or deny the request through the exchangeDenyReason message.

The exchangeData message has the following mandatory elements

- Payload
  - CataloguePublication with all selected catalogues
- elements shown in the object form:

Figure 27 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            <CataloguePublication>
                  …
            </CataloguePublication>
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The exchangeDenyReason message has the following mandatory elements.

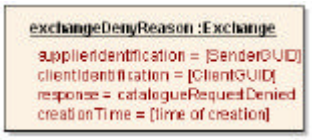

Figure 28 – Exchange Deny Reason Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>catalogueRequestDenied</response>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

5.2.4. Filter Publication Service

5.2.4.1. Filter Operation Acknowledgement

To request filter operations, the client should use the methodology described by the FilterPublication package. To acknowledge the creation or deletion of filters on the supplier system the filterOperationApproved flag is set corresponding to the action taken.

Request Filter Publication

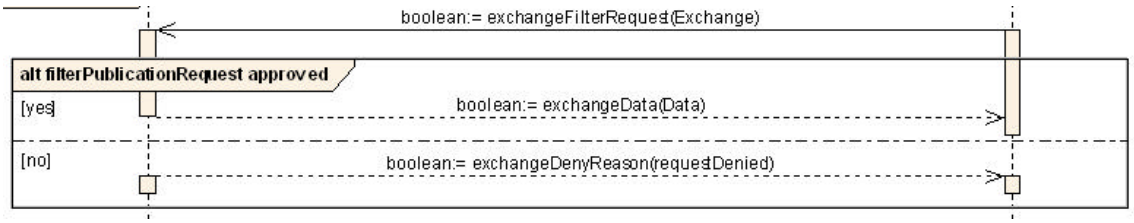

Figure 29 – Request Filter Publication

The possibility to publish filters is optional. Where used the Client sends a exchangeFilterRequest message which can be either approved by sending a exchangeData message or denied by using the exchangeDenyReason message. It is also possible to send customer defined filters from the client system to the DATEX 2 System.

If one of the filters has been changed the element changedFlag shall be used to signal this. In such a case, only the Exchange class with its mandatory elements and the changedFlag set to "filter" is sent. After receiving this information, the client can request the updated filter.

The filterPublicationRequest has to have the following mandatory elements

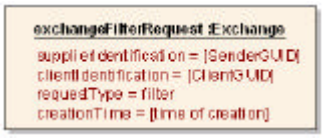

Figure 30 – Exchange Filter Request Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
        <Exchange>
                <supplierIdentification>SenderGUID</supplierIdentification>
                <clientIdentification>ClientGUID</clientIdentification>
                <requestType>filter</requestType>
                <creationTime>2004-08-14T11:06:28</creationTime>
        </Exchange>
</D2LogicalModel>
```

A message with the following mandatory elements has to be sent:

- Payload
  - FilterPublication with all available filters for the Client
- Elements shown in the object form:

Figure 31 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            <FilterPublication>
                  …
            </FilterPublication>
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The requestDenied message has the following mandatory elements.

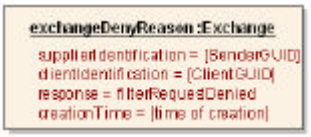

Figure 32 – Exchange Deny Reason Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>filterRequestDenied</response>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

5.2.4.2. Create Customer Defined Filters

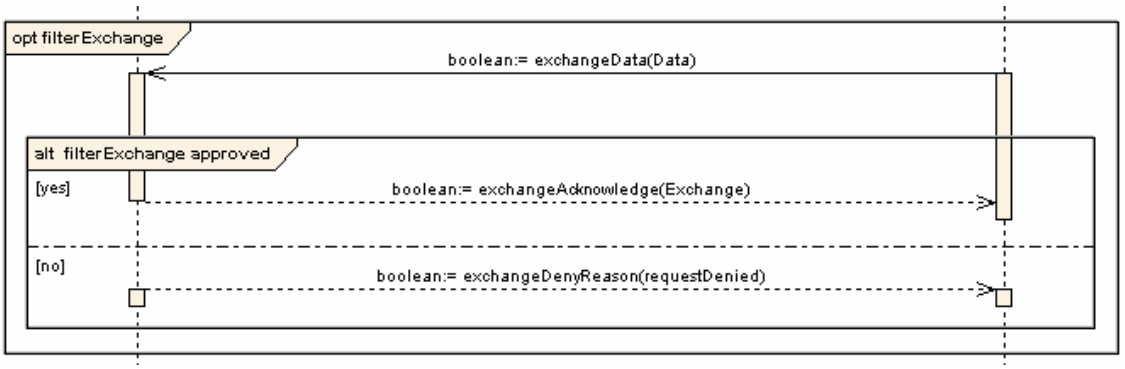

Figure 33 – Message "ClientFilterPublication"

These two optional messages enable a Client to send a filter defined by the Client to the DATEX 2 System of the supplier. The Client sends the filters with the “exchangeData(Data)” message to the DATEX 2 System where the filters are stored and returns a summary of this operation to Client.

The first message in the figure above has the following mandatory elements.

- Payload
  - FilterPublication with all customer defined filters
- Elements shown in the object form:

Figure 34 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            <FilterPublication>
                  …
            </FilterPublication>
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The second message in the figure above has the following mandatory elements.

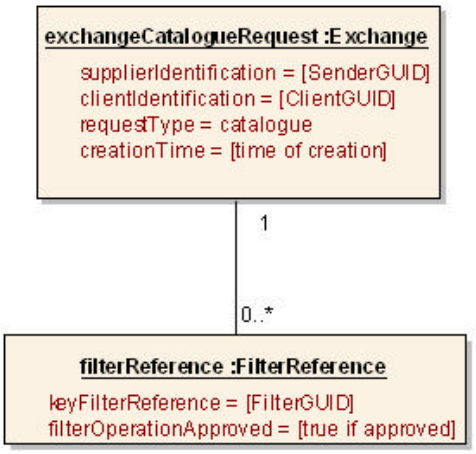

Figure 35 – Exchange Acknowledge Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <FilterReference>
                  <keyFilterReference>FilterGUID</filterReference>
                  <filterOperationApproved>true</filterOperationApproved>
            </FilterReference>
                  <filterReference>
                        <keyFilterReference>FilterGUID</filterReference>
                  <filterOperationApproved>false</filterOperationApproved>
            </FilterReference>
      </Exchange>
</D2LogicalModel>
```

The example above shows the responses to two transmitted customer defined filters where the first is approved and the second is denied.

Where it is not possible to import customer defined filters the exchangeDenyReason is used.

5.2.4.3. Delete Customer Defined Filters

These two optional messages provide the mechanism of deleting a transmitted customer defined filter within the DATEX 2 System. To delete customer defined filters the first message of the figure above is used with the following mandatory elements

Figure 36 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <creationTime>2004-08-14T11:06:28</creationTime>
            <FilterReference>
                  <filterReference>FilterGUID</filterReference>
                  <delete>true</delete>
            </FilterReference>
      </Exchange>
</D2LogicalModel>
```

The response to a deletion request is sending back the second message of the figure above with the following mandatory elements.

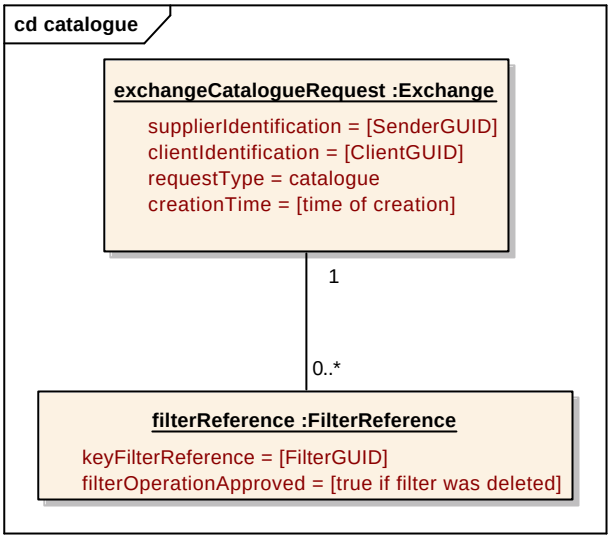

Figure 37 – Exchange Acknowledge Reference

5.3. Operating Mode 1 – Supplier Push On Occurrence

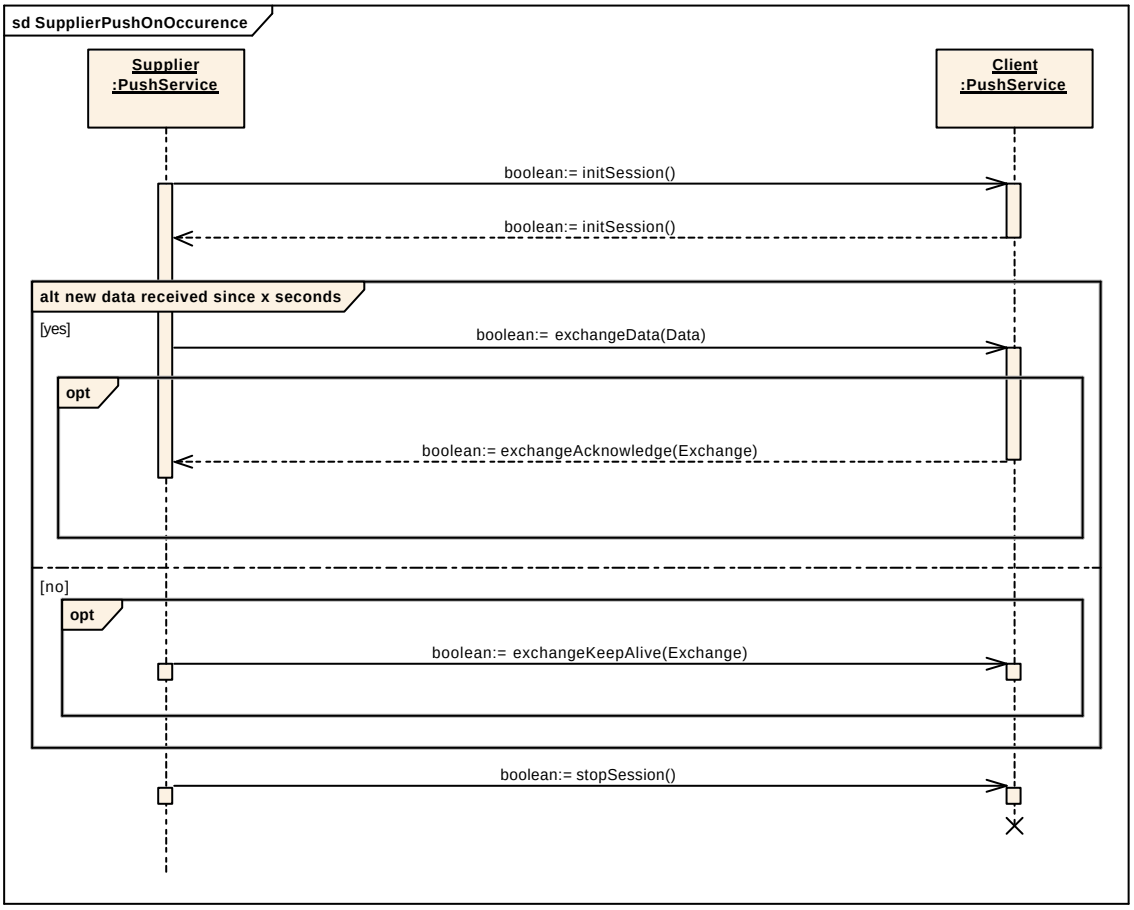

Figure 38 – Sequence Diagram "SupplierPushOnOccurence"

The figure above shows the Operating Mode 1 “SupplierPushOnOccurence”.

This Operating Mode uses the same session management as the subscription mechanism. The session management can be enhanced by the use of keep-alive functionality, monitoring and preserving the link between DATEX 2 System and the Client. If no data has been available for transmission to the client for a stipulated period and thus none has been sent to the Client, a keep-alive message will be generated and sent.

Without use of catalogues and filters it is up to the supplier to define what is sent.

The exchangeKeepAlive message has the following mandatory elements.

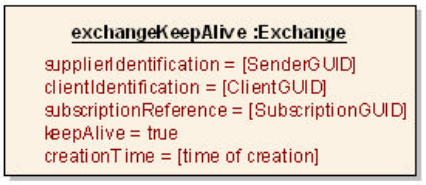

Figure 39 – Exchange Keep-alive Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <keepAlive>true</keepAlive>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The exchangeData message has to have the following mandatory elements.

- Payload
- Elements shown in the object form:

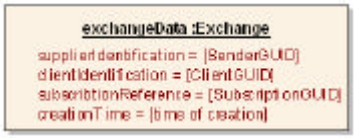

Figure 40 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            …
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

An optional exchangeAcknowledge can be sent by the Client in order to enhance the monitoring of the connection. The exchangeAcknowledge message has the following mandatory elements

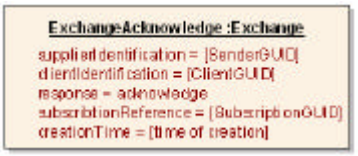

Figure 41 – Exchange Acknowledge Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>acknowledge</response>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

5.4. Operating Mode 2 – Supplier Push Periodic

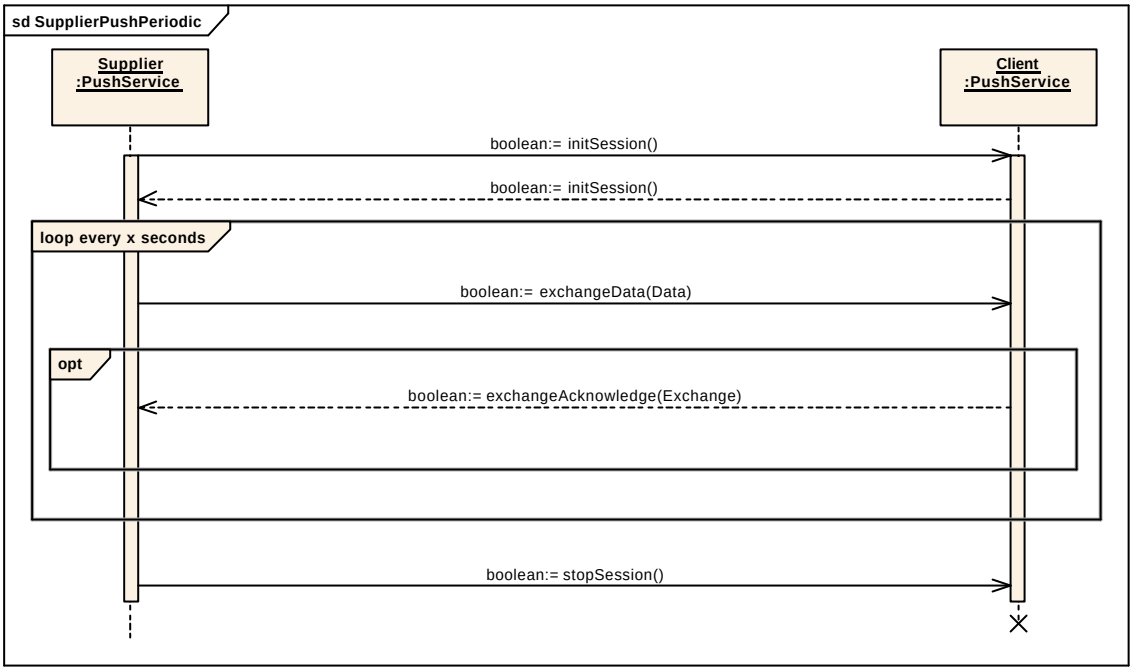

Figure 42 – Sequence Diagram "SupplierPushPeriodic"

The above figure illustrates the Operating Mode 2 “SupplierPushPeriodic”.

This Operating Mode uses the general session management functionalities.

> Without use of catalogues and filters it is up to the supplier to define what is sent.

The exchangeData message has to have the following mandatory elements.

- Payload
- Elements shown in the object form:

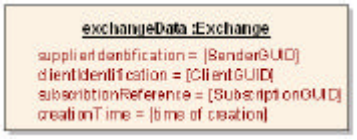

Figure 43 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            …
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

An optional exchangeAcknowledge can be sent by the Client to enhance the monitoring of the connection. The exchangeAcknowledge message has the following mandatory elements

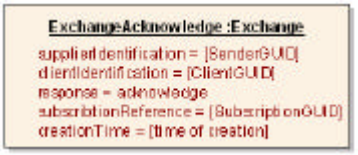

Figure 44 – Exchange Acknowledge Reference

The following XML code is provided as an e xample:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>acknowledge</response>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

This Operating Mode does not need keep-alive functionality because its cyclic data transmission given by the nature of this Operating Mode.

5.5. Operating Mode 3 – Client Pull

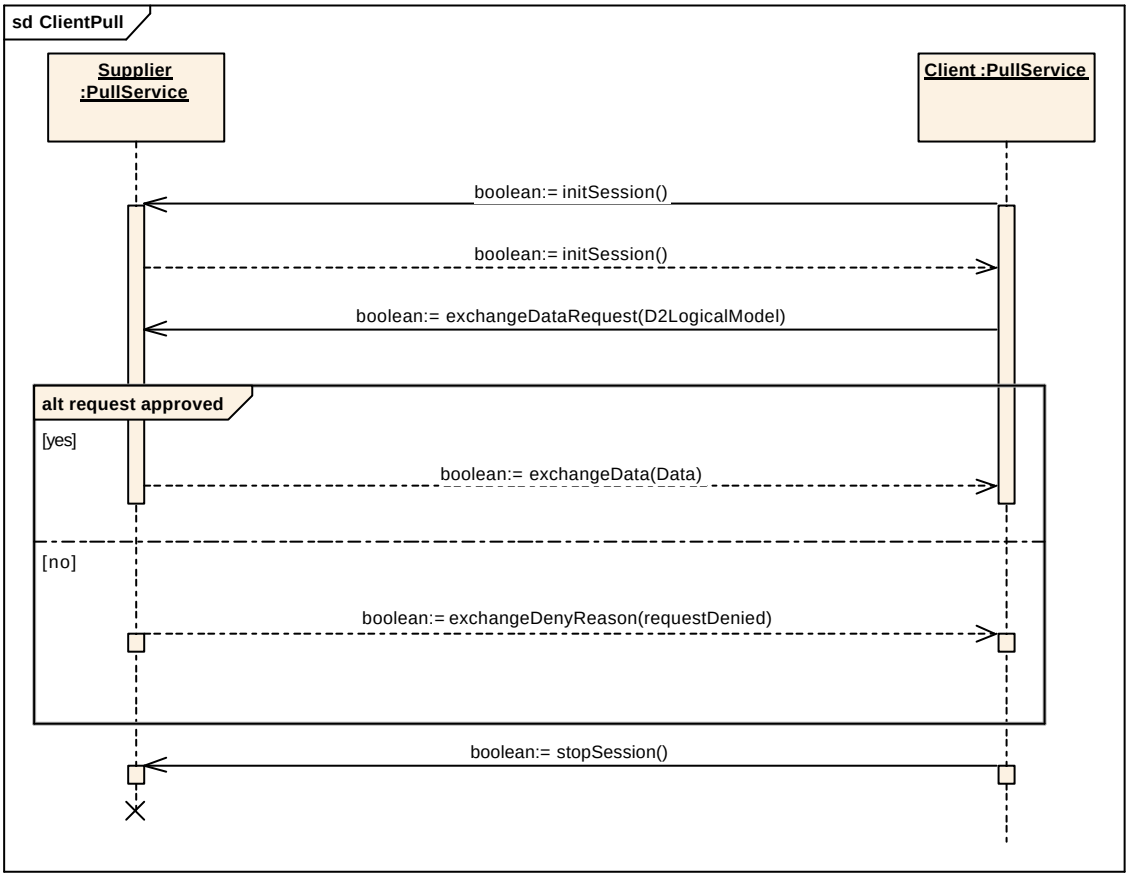

Figure 45 – Sequence Diagram "ClientPull"

The figure above illustrates the Operating Mode 3 “ClientPull”.

This Operating Mode uses the general session management functionalities.

> Without use of catalogues and filters it is up to the supplier to define what is sent.

The exchangeDataRequest message has to have the following mandatory elements.

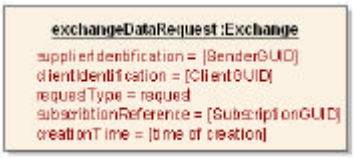

Figure 46 – Exchange Data Request Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
```

```
            <clientIdentification>ClientGUID</clientIdentification>
            <requestType>request</requestType>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

If this request is denied the exchangeDenyReason message will be sent to the Client. This message has the following mandatory elements.

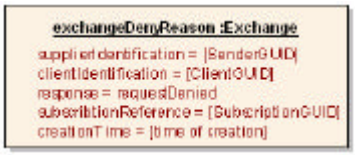

Figure 47 – Exchange Deny Reason Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <response>requestDenied</response>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

The exchangeData message has to have the following mandatory elements.

- Payload
- Elements shown in the object form:

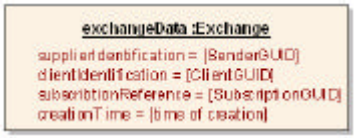

Figure 48 – Exchange Data Reference

The following XML code is provided as an example:

```
<D2LogicalModel>
      <Payload>
            …
      </Payload>
      <Exchange>
            <supplierIdentification>SenderGUID</supplierIdentification>
            <clientIdentification>ClientGUID</clientIdentification>
            <subscriptionReference>SubscriptionGUID</subscriptionReference>
            <creationTime>2004-08-14T11:06:28</creationTime>
      </Exchange>
</D2LogicalModel>
```

ANNEX

6. Annex

6.1. Interchange Agreement

The interchange agreement is a document that describes bilateral agreements between Supplier and Client needed for the exchange of DATEX2 information. The interchange agreement should deal with all the information that is needed for the data exchange to take place. The description assumes that the information exchange follows the basic principles of DATEX2. Specifically it is assumed that two parties are involved: a Supplier providing content and a Client receiving content.

6.1.1. General

Time period throughout which the overall agreement is valid

The agreement shall state the time period for which the agreement is valid. It should define the date of commencement and may indicate the date of its ending.

Rules for terminating the agreement before the expiry time of the agreement

The agreement should specify the conditions under which the contract can be terminated prior to an expiry date.

Company/authority names, contact addresses, telephone, fax and E-mail details

The two parties shall provide the company and the authorities´ names, address and the operational contact with all related communication data such as telephone, fax numbers and email addresses.

6.1.2. Access

Address of the Client

The agreement should mention the network address of the client which is to be used for setting up the communication between the parties.

Address of the Supplier

The agreement can mention the network address of the supplier that will be used for setting up the communication between the parties.

Supplier user name

The agreement should stipulate the user name that will be adopted by the supplier.

Client user name

The agreement should stipulate the user name that will be adopted by the supplier.

Password for Supplier to connect to Client

If the parties decide to use a password, this should be specified. If other means of access control are used these should be specified.

Password for Client to connect to Supplier

If the parties decide to use a password, this should be specified. If other means of access control are used they should be specified.

6.1.3. Data exchange

Choice of public network for exchange of data and configuration

All technical information needed for the exchange of data and configuration between the parties should be described (Example: ISDN, special line … and router parameters...)

6.1.4. Rights and obligations

Description of the quality of the data provided by Supplier.

Description of actions by Supplier in connection with the Supplier Server being down:

- actions when Supplier Server is down for planned maintenance,
- actions when Supplier Server is down following a system failure,
- actions when Supplier server starts up.

Description of actions by Client in connection with the Client system being down:

- actions when Client system is down for planned maintenance,
- actions when Client system is down following a system failure,
- actions when Client system starts up.

Description of payment mechanism for the service, if any.

Statement on liability in case of transmission of incorrect information.

Rules for the further dissemination by the Client of information received using DATEX2.

6.2. Table Of Figures

Figure 1 – Analysis Diagram "BasicSystemOverview" 4
Figure 2 – Analysis Diagram "SubSystem" 5
Figure 3 – Use Case Diagram "SystemOverview" 6
Figure 4 – Use Case Diagram "HandleSubscription" 7
Figure 5 – Class Diagram "Exchange" 9
Figure 6 – Class Diagram "ExchangeEnum" 10
Figure 7 – Class Diagram “ExchangeServices” 12
Figure 8 – Sequence Diagram "Subscription" 18
Figure 9 – Sequence Diagram Part “initSession” 18
Figure 10 – Sequence Diagram Part “stopSession” 18
Figure 11 – Sequence Diagram Part “subscriptionRequest” 19
Figure 12 – Create Subscription 19
Figure 13 – Exchange Deny Reason Reference 20
Figure 14 – Exchange Subscription Reference 20
Figure 15 – Edit Subscription Mechanism 21
Figure 16 – Exchange Data Reference 21
Figure 17 – Exchange Data Reference 22
Figure 18 – Exchange Data Reference 23
Figure 19 – Exchange Acknowledge Reference 24
Figure 20 – Exchange Deny Reason Reference 24
Figure 21 – Exchange Data Reference 25
Figure 22 – Catalogue Publication Service 26
Figure 23 – Exchange Catalogue Request Reference 26
Figure 24 – Exchange Data Reference 27
Figure 25 – Exchange Deny Reason Reference 27
Figure 26 – Exchange Catalogue Request Reference 27
Figure 27 – Exchange Data Reference 28
Figure 28 – Exchange Deny Reason Reference 28
Figure 29 – Request Filter Publication 29
Figure 30 – Exchange Filter Request Reference 29
Figure 31 – Exchange Data Reference 29
Figure 32 – Exchange Deny Reason Reference 30
Figure 33 – Message "ClientFilterPublication" 30
Figure 34 – Exchange Data Reference 30
Figure 35 – Exchange Acknowledge Reference 31
Figure 36 – Exchange Data Reference 31
Figure 37 – Exchange Acknowledge Reference 32
Figure 38 – Sequence Diagram "SupplierPushOnOccurence" 32
Figure 39 – Exchange Keep-alive Reference 33
Figure 40 – Exchange Data Reference 33
Figure 41 – Exchange Acknowledge Reference 33
Figure 42 – Sequence Diagram "SupplierPushPeriodic" 34
Figure 43 – Exchange Data Reference 34
Figure 44 – Exchange Acknowledge Reference 34
Figure 45 – Sequence Diagram "ClientPull" 35
Figure 46 – Exchange Data Request Reference 35
Figure 47 – Exchange Deny Reason Reference 36
Figure 48 – Exchange Data Reference 36