VDA | Verband der Automobilindustrie

Quality Management in the Automotive Industry

QDX data exchange requirements

QDX specifications - Quality Data eXchange V2.0

2nd revised edition 2011

QDX data exchange requirements

QDX specifications - Quality Data eXchange V2.0

2nd revised edition 2011

Verband der Automobilindustrie e.V. (VDA)

Published 10/2011

English edition 09/2012

Verband der Automobilindustrie e. V. (VDA)
Qualitäts Management Center (QMC)
10117 Berlin
Behrenstraße 35

Germany

Production :

Henrich Druck + Medien GmbH
60528 Frankfurt am Main
Schwanheimer Str. 110

Germany

Non-committal VDA recommendation regarding standards

The Verband der Automobilindustrie (Automotive Industry Association - VDA) proposes that its members apply the following recommendations regarding standards when establishing and maintaining QM systems.

Exclusion of responsibility

This VDA document is a recommendation which is free for anyone to use. Anyone using it must ensure that it is applied correctly in each individual case.

This VDA document takes account of the latest state of technology at the time it is issued. The application of the VDA recommendations does not in any way relieve the user of his own responsibility for the use of the document. To this extent, the user applies the document at his own risk. The VDA and those involved in drawing up the VDA recommendations decline all liability in any circumstances.

Anyone using these VDA recommendations detecting incorrect information or the possibility of incorrect arrangements is asked to advise the VDA without delay, so that any deficiencies can be eliminated.

References to standards

The individual standards referred to by their DIN number and their dates of issue are quoted with the permission of the DIN (German Standards Institute). It is essential to use the latest issue of the standards, which are available from Beuth Verlag GmbH, 10772 Berlin, Germany.

Copyright

Translations

This document will also appear in other languages. Please contact the VDAQMC for the latest position.

We thank the participating organisations and their employees for their contributions in the compilation of this document. The following companies were involved in drawing up the volume (in alphabetical order) :

- AUDI AG
- Axway
- Daimler AG
- Harman/Becker Automotive Systems GmbH
- iPoint-systems GmbH
- Mieschke Hofmann und Partner Gesellschaft für Management- und IT-Beratung mbH
- msg systems ag
- SEEBURGER AG
- SupplyOn AG
- T-Systems AG
- Volkswagen AG

Our thanks also go to all who have given us encouragement and assistance in generating and improving the document.

Berlin, May 2011

Verband der Automobilindustrie e.V. (VDA)

Contents **Page**

Management Summary (abstract)

After the contents of XML files to be transmitted using QDX had been described in detail, it was found that direct communication (that is, the direct exchange of data) between the IT departments of two commercial partners can function only if the way in which these documents are exchanged is also specified.

With the objective of establishing a uniform and central requirement specification, a working group made up of personnel from the organisations listed at the end of the preface to this document has drawn up a guideline recommendation.

This present guideline (in the form of a technical specification) therefore forms the technical basis for the standardisation of quality data exchange envisaged by the use of QDX. Using this document it will be possible to create QDX interfaces, across which communication can take place between all commercial partners without further technical modification, provided all parties have implemented an interface which complies with this present guideline.

From the wide range of possible paths of communication, more modern communication using WebServices has emerged as the only other alternative to the conventional paths using OFTP and EDI.

In the course of work on this present guideline it became clear that further QDX documents are required, containing information on the actual data transmission. As a result the QDX standard had to be expanded to cover this aspect. The additional QDX documents are described within the framework of this present document.

In order to inspect the current QDX specification documents you must first register. To do so, please send an e-mail with your company and contact details to ide@vda-qmc.de. Once you have registered, you will receive an e-mail within 2 working days,

containing a link which will enable you to download the specification documents.
Registration and downloading are free and do not involve any further commitment of any kind.

The actual QDX messages are "packed" in envelopes (more or less as envelopes for letters), so that incoming messages can be directed to the correct application via the routing system and so that attachments can be transmitted.

The same SOAP and QDX envelopes are used for both paths of communication. Descriptions of the SOAP and QDX envelopes specified by the working group are provided in this present guideline.

Finally it should be pointed out that, even with this present guideline, the commercial partners will need to define their own, additional requirements, so that communication functions effectively (for example, what alternative communication is used, what alternatives are possible and which of the optional QDX data fields must be transmitted without fail, with what contents and with what maximum data field length, etc.)

1 Introduction

This present document describes the technical exchange of QDX documents. Its primary objective is the implementation of uniform QDX interfaces in all organisations. For this purpose, two communication procedures (OFTP and WebService) have been selected, which should be used for exchanging QDX documents. In addition, this document aims to make the implementation of a QDX interface more transparent and therefore more simple.

The document is structured as follows :

- Section 2 is an introduction to the subject. It explains the types of communication and examines the questions of routing and procedure reference. This basic understanding is essential in order to follow the definition of communication procedures. This section is set out in general terms and can be used for all QDX processes; only the examples refer to the complaints process and initial sample submissions.
- Section 3 describes the two communication procedures which have been defined for exchanging QDX documents. Other communication procedures should not be used. To ensure better understanding it is advisable to study Section 2 beforehand. This section describes only the complaints process by reference to the QDX documents *QDXComplaint* and *QDXReport8D*.
- Section 4 describes the "envelope" in which the QDX documents are transported. The envelope contains the information required for the routing. The section describes only the QDX documents *QDXComplaint* and *QDXReport8D* (and the documents which are also required for these two to be exchanged).
- Section 5 contains the WSDL definition, XML schematics and example files. The WSDL definition and the XML schematics are intended to make the implementation of a Webservice more transparent and therefore simpler. The example files help in the creation of the QDX documents.

1.1 Limitations

The QDX integration process and the associated technical exchange of QDX documents are described within the framework of this present document. A description of the commercial processes is not the purpose of this document. This specification covers primarily the complaints process and the associated QDX documents *QDXComplaint* and *QDXReport8D*. However, it is easily possible to expand the document to cover the other quality processes and their QDX documents.

The specification is based on QDX Version 2.0. The QDX documents described in Section 4.4 have been included in QDX Version 2.0 and also the changes to references for attachments – as described in Section 4.5. However, in principle it is possible to implement this specification using QDX Version 1.2, Here it should be noted that the QDX documents in Section 4.4 did not yet exist in QDX Version 1.2. In addition, a Workaround is required for referencing attachments[1].

The subjects of signing, security, monitoring and prioritisation are not covered by the specification. It is intended to cover signing and security at a later date; at present, however, there is no demand from the commercial process. Likewise, there is no demand for prioritisation. On the other hand, monitoring the exchange of data is a task for the systems which exchange QDX documents. Because of this, no description is provided here.

[1] The field "URL" must be used for the Content-ID.

1.2 Outlook

This document covers only the complaints process. It is intended for use in practice in the first instance. In a second version, the experience gained in practice will be used to revise the definitions as necessary.

However, this specification can be adapted without difficulty from the other quality processes described in QDX. It is merely necessary to adapt the structure of the QDX documents and the functionality of the Webservice methods to the references of the relevant QDX documents. However, it is important to note who the initiator of the process is (see Section 2.1).

The life-cycle of IT technologies and IT standards is becoming ever shorter. The aim of this present document is to describe the communication technologies which are current in the automotive sector. Because of this, the specification can be expanded to cover new communication technologies – provided that the number of communication technologies described is kept as low as possible. To make it easier to expand to cover other communication technologies, Section 2 does not describe any specific communication technology. Instead, the two possible types of communication covered by this present document are described.

2 QDX integration process

The QDX integration process is designed to link the customer's and supplier's CAQ systems together so that no media breaks occur when information is exchanged between them. The QDX integration process therefore describes the communication between the CAQ systems of the commercial partners.

The first step is to determine the initiator of the individual commercial process (that is, who sends the initial message), because this determines which of the commercial partners must save information for the routing or the mapping covering a transmission[2].

Initiator of the commercial process : The initiator initiates the process and waits fro a response from the other party. The initiator may be either the customer or the supplier :

- **The customer as initiator :** If the customer is the initiator of the process (for example, in the complaint process) the supplier must send a response (an 8D report) covering the customer's message (the complaint) to the customer's system. The supplier's response must include the initial message reference and the identification of the customer's system[2].
- **The supplier as initiator :** If the supplier is the initiator (for example, an initial sample submission) the customer must provide a response (production process & product release – PPF) to the message (a PPF report) to the supplier's system. The customer's response must include the initial message reference and the identification of the supplier's system[2].

[2] The information to be provided by a commercial partner when exchanging data is covered in the Sections 2.1 (Routing) and 2.1.2 (Message reference).

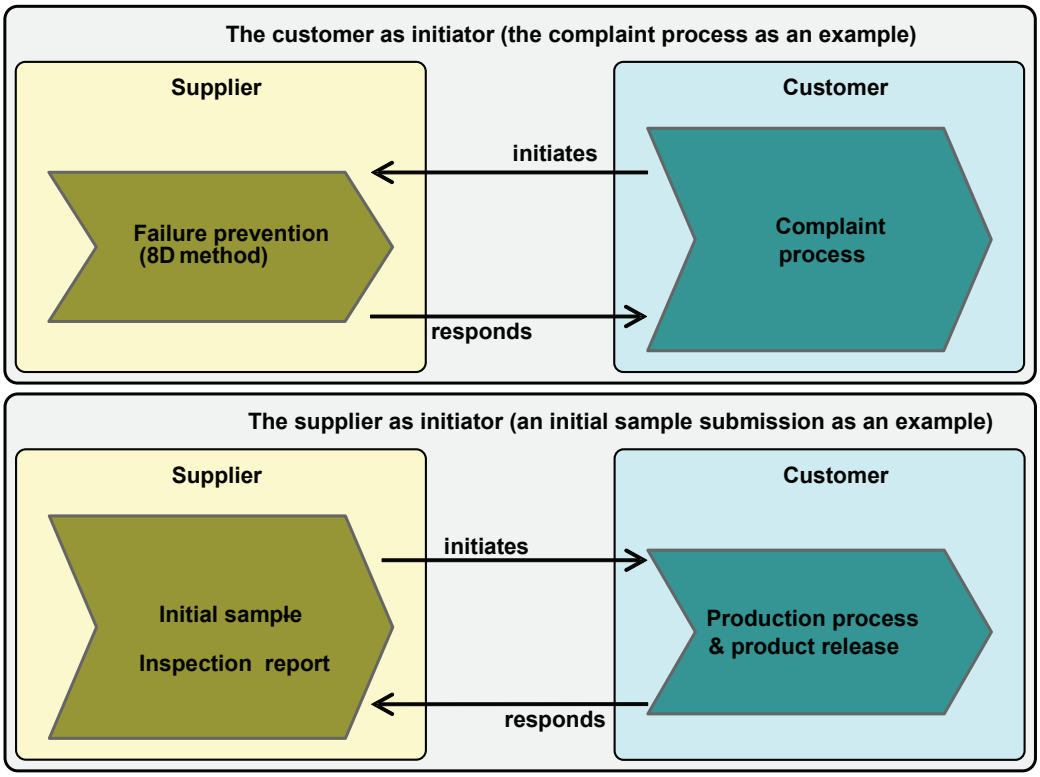

Fig. 2-1 : Who initiates the commercial process

2.1 Routing

Communication between the CAQ systems of the two commercial partners is very simple if both CAQ systems communicate directly with each other. However, so-called routing systems are used in large organisations to control communication between internal and external systems. The consequence of this is that communication becomes much more complex. For example, these routing systems need information, on the basis of which they can forward the QDX documents to their correct destination.

Possible scenarios are described briefly in the following pages.

Scenario 1: Both CAQ systems communicate directly with each other. This is the simplest scenario, as no further information is required.

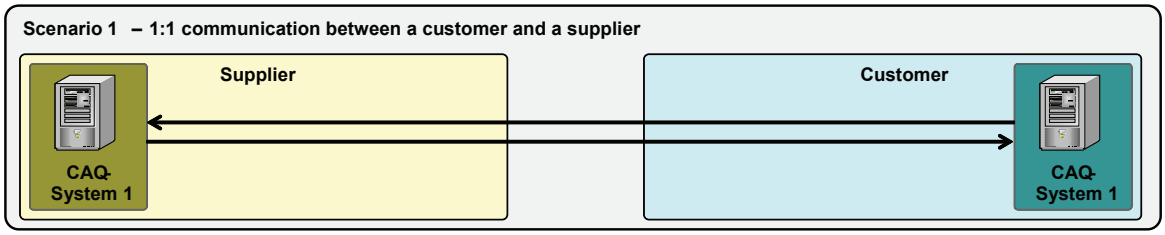

Fig. 2-2: Scenario 1 - 1:1 communication between customer & supplier

Scenario 2: On the customer's side, communication between internal and external systems is controlled by a routing system. The supplier number is needed in order to send QDX documents to the supplier. In order to forward incoming QDX documents to the correct CAQ system, the routing system needs the location description (the customer number) as a minimum and, as a general rule, a system identification.

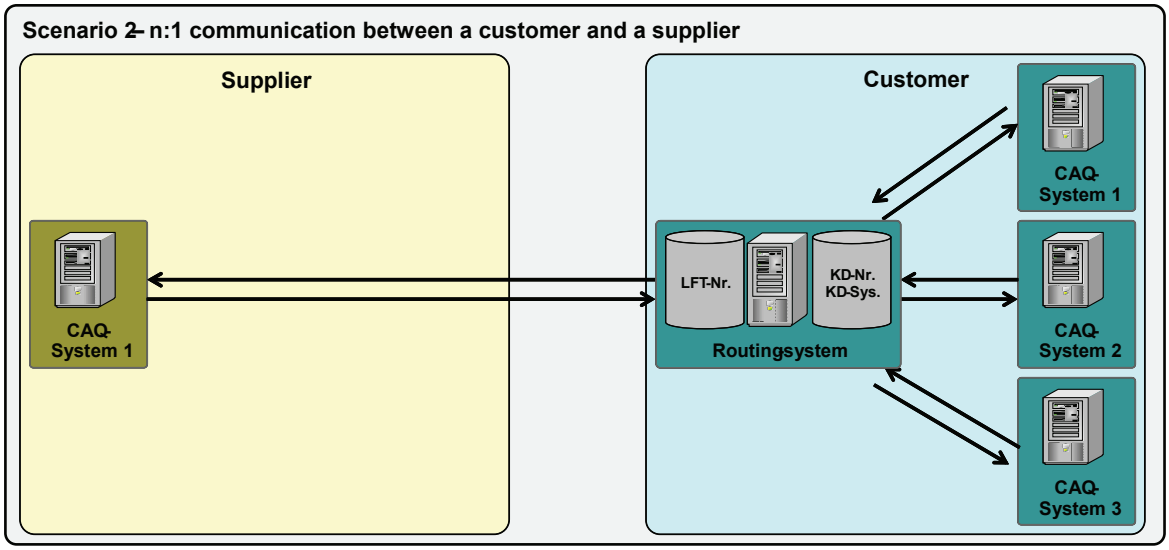

Fig. 2-3: Scenario 2 - n:1 communication between customer & supplier

Scenario 3: This is similar to Scenario 2. On the supplier's side, communication between internal and external systems is controlled by a routing system. So that incoming QDX documents can be forwarded to the correct CAQ system, the supplier number is required as a minimum and maybe a system identification. Only

the location number (customer number) is required in order to send QDX documents from the supplier to the customer.

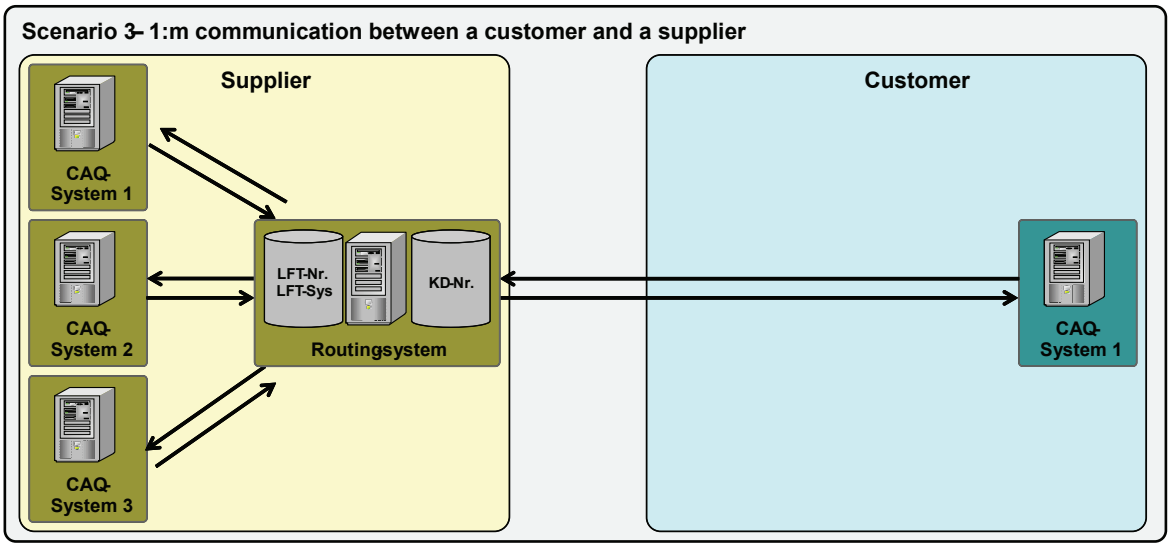

Fig. 2-4: Scenario 3 - 1:m communication between customer & supplier

Scenario 4: This scenario represents a mix of scenarios 2 und 3 – both commercial partners have a routing system for controlling communication. For outgoing documents the reference number of the other party (the customer number or supplier number) is sufficient. For incoming documents a system identification is usually also required.

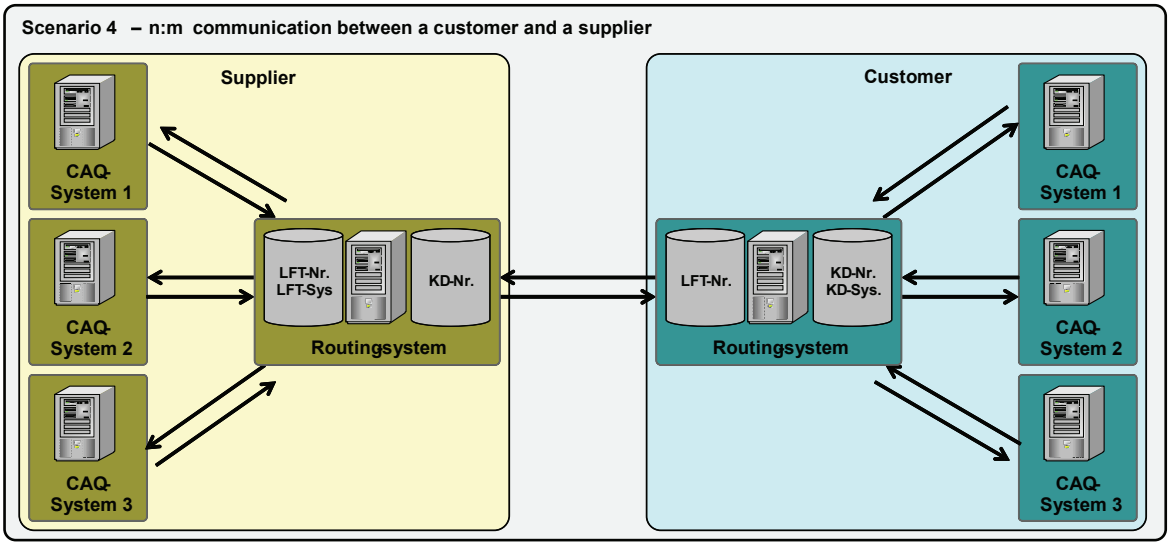

Fig. 2-5: Scenario 4 - n:m communication between customer & supplier

Because of the complexity of some scenarios, additional information must be supplied when data are transmitted, which the routing systems use in order to decide which is the correct final receiving system for the data The commercial partners need to provide different information, depending on who initiates the message.

Table 2-1 defines the information required by the routing system. How this information is used is described in the following pages.

Reference	Description
KD-Nr.	**Customer number :** the customer identification. This is the number which the customer has registered for himself.
KD-Sys.	**Customer system :** the identification of the customer's CAQ system. The customer can select any reference for this.
LFT-Nr.	**Supplier number :** the supplier identification. This is the number which the customer has allocated to the supplier.
LFT-Sys.	**Supplier system :** the identification of the supplier's CAQ system. The supplier can select any reference for this.

Table 2-1: Description of routing information

2.1.1 The customer as initiator

If the customer is the initiator of the commercial process, the supplier number is sufficient in order to direct the initial message onwards in the customer's routing system. However, the routing system on the supplier's side may require additional information – for example, if there are several CAQ systems operating in a location. So that the customer does not have to keep up-dating a numbering system for all suppliers, it was decided that the customer number, the customer's system number and the supplier number must be sufficient for the supplier's routing system[3].

[3] However, not all the information mustbe used for the routing.

Otherwise, the supplier is responsible for ensuring that the message is forwarded to the correct system.

If the supplier, in his role as the commercial partner, sends back a response, the customer number is sufficient to ensure that the message is forwarded correctly in his routing system. However, the routing system on the customer's side may require additional information, because, here again, a single location may have several CAQ systems in operation. Unlike the previous case, however, the operation is a response to a message from the customer and the customer has already sent information to the supplier in this connection. When sending the initial message, the customer has the option of stating the system number of his source system. The supplier must save this and quote it in his response. In this way, the routing system on the customer's side receives all the information required to direct the response onwards to the correct CAQ system.

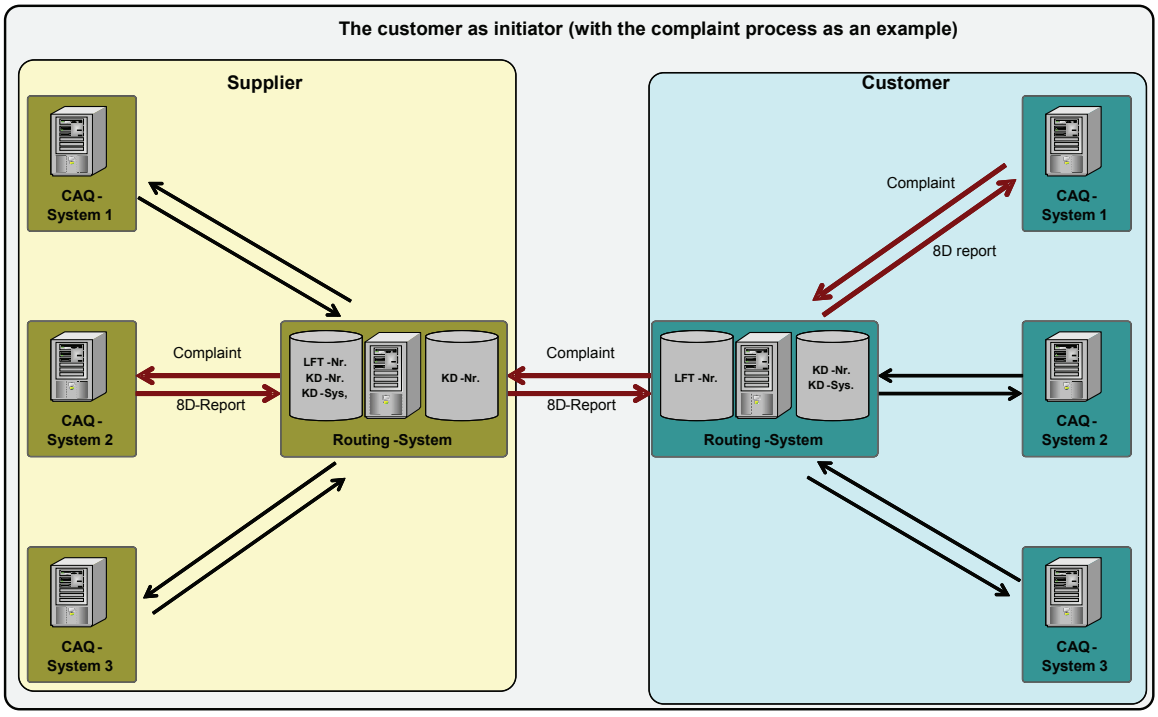

Fig. 2-6: Illustration of routing with the customer as initiator[4]

Reference	Description
KD-Nr.	**Customer number :** the customer identification. This is the number which the customer has registered for himself.
KD-Sys.	**Customer system :** the identification of the customer's CAQ system. The customer can select any reference for this. The customer can use this if he wishes. If he does so, the supplier must include this information in his response.
LFT-Nr.	**Supplier number :** the supplier identification. This is the number which the customer has allocated to the supplier.

Table 2-2: Usable routing information

[4] The data sink to the initiating CAQ system describes the routing information which the routing system needs for routing a message to the correct CAQ The data sink to the other party's system, on the other hand, describes the routing information needed for routing from his routing system to that of his commercial partner.

2.1.2 The supplier as initiator

If the supplier is the initiator of the commercial process, the customer number is sufficient for the message to be forwarded correctly in the supplier's routing system. However, the routing system on the customer's side may require additional information – for example, if there are several CAQ systems operating in a location. So that the customer does not have to keep up-dating a numbering system for all suppliers, it was decided that the supplier need include only the customer number and the customer's system number in order for the message to be forwarded to the correct CAQ system.

If the customer in his role as commercial partner sends back a response, the supplier number is sufficient for the message to be forwarded correctly in the customer's routing system. However, the routing system on the supplier's side may require additional information – for example, if there are several CAQ systems operating in a location. However, unlike the previous case, the message is a response to a message initiated by the supplier. When sending the initial message, the supplier is able to include the system identification of the source system. The customer must save this and include it in his response. In this way, the supplier's routing system receives all the information required to forward the response to the correct CAQ system.

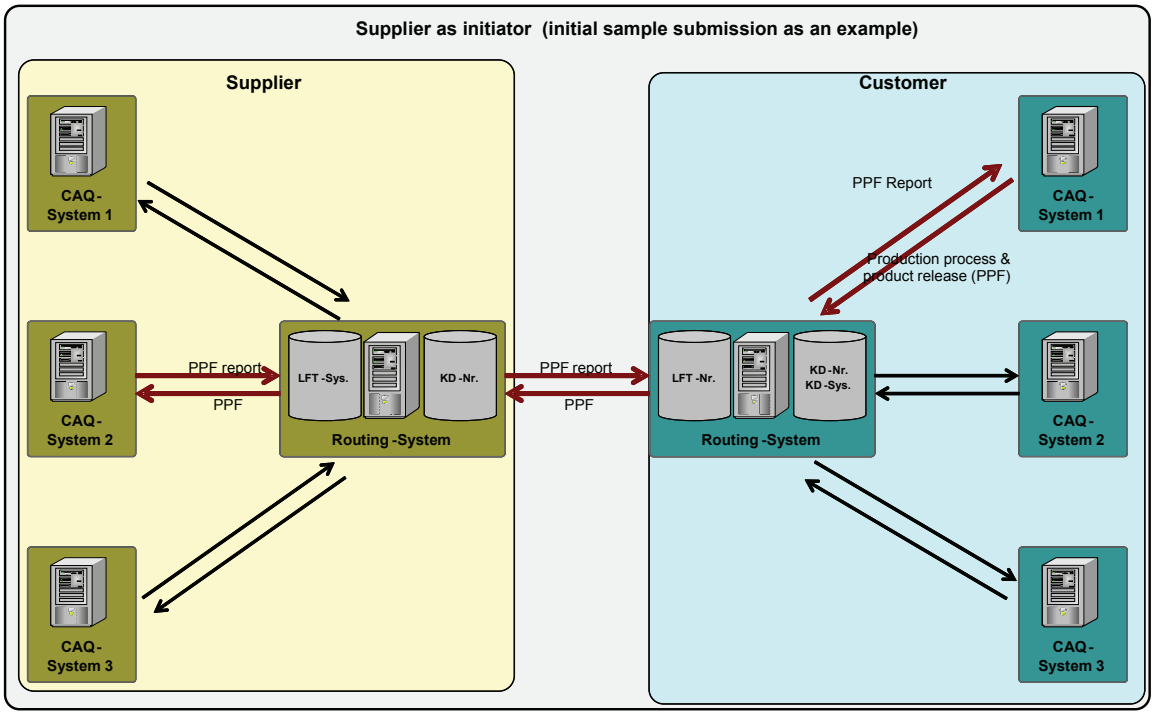

Fig. 2-7: Illustration of the routing with supplier as initiator[4]

Reference	Description
KD-Nr.	**Customer number :** the customer identification. This is the number which the customer has registered for himself.
LFT-Nr.	**Supplier number :** the supplier identification. This is the number which the customer has allocated to the supplier.
LFT-Sys.	**Supplier system :** the identification of the supplier's CAQ system. The supplier can select any reference for this. The supplier can use this if he wishes. If he does so, the customer must include this information in his response.

Table 2-3: Description of routing information

2.2 Message reference

Within the framework of quality processes supported by QDX, various QDX documents are exchanged between the commercial partners. This generally involves an initial message which causes a process to begin at the other party's end, as well as a response, which closes the process the process which was launched. So that the initiator's receiving system can allocate the response to the correct initial message, the other party must provide information which will enable the initiator to do so.

Table 2-4 describes the information which the receiving party must save and include in his response to the initiator. The first two pieces of information are generally applicable and should therefore always be saved. In addition, it may be necessary to save further information on a process, where the process is made up of several sub-processes – such as, for example, the complaint process or the initial sample submission process.

DocumentID	This identifies the document sent by the initiator (complaint, PPF report, etc.). the document ID to be used is the one which the initiator has stated in the header of the QDX document.
RevisionID available as an option)	This identifies the version of the document which is sent. However, this application is optional and is used to support the communication. The version number to be used is the one which the initiator has stated in the header of the QDX document
RevisionDateTime	The time-stamp of the version, together with the DocumentID provides an unambiguous identification of the document which has been sent. The version time stamp makes it possible to identify clearly the version levels of the documents. The time stamp to be used is the one which the initiator has stated in the header of the QDX document under RevisionDateTime.
ComplaintItemID (complaint process)	This identifies the individual complaint issued by the customer and is used to differentiate between the items in a test/inspection report. (because the test/inspection report number is usually stated as the DocumentID). The ID to be used is the one which the initiator has allocated to the ComplaintItem
BuyerProductItemID (initial sample submission)	This identifies the individual product for the customer (the part number) and is used to differentiate between the items in a PPF report (a PPF report can cover several items). However, the part number stated must always originate from the customer's numbering system.

Table 2-4: Message references to be stored by the commercial partner

The elements described above, used to identify the commercial document, must be quotes in the response in the header of the QDX document under "ReferenceDocument“. If this information is not included in the response, the initiator will not be able to allocate the response to the relevant message.

In addition, however, the initiator of the commercial process must also save information in the response from the other party. The main reason for this is to be able to confirm the successful processing of a response which has been received. This information is set out in detail in Table 2-5.

DocumentID	This identifies the document sent by the other party (8D report, PPF, etc.). The DocumentID to be used is the one which the other party has stated in the header to the QDX document.
RevisionID available as an option)	This identifies the version of the document which is sent. However, this application is optional and is used to support the communication. The version number to be used is the one which the initiator has stated in the header of the QDX document
RevisionDateTime	The time-stamp of the version, together with the DocumentID provides an unambiguous identification of the document which has been sent. The version time stamp makes it possible to identify clearly the version levels of the documents. The time stamp to be used is the one which the initiator has stated in the header of the QDX document under RevisionDateTime.

Table 2-5: Message references to be stored by the initiator

2.3 Types of communication

When considering communication, the first and most important point is who initiates the communication. In this a differentiation is made between two types of communication :

- **Active communication :** The initiator of the communication is the party which wishes to transmit information to the other party. Each party sends its information actively to the other and is therefore responsible for the structure of the communication.

- **Passive communication :** It is also possible, however, for the responsibility to be borne by only one party, so that only one of the two commercial parties is responsible for the establishing the communication structure. The other party always takes a passive role. However, only the customer can take the passive role and the supplier is always in the active role. For the supplier, this means that he receives information from the customer only if he explicitly "asks" for it. If he does not "ask" he receives no new information.

Before the integration of a commercial process the customer must specify if he will take an active role in establishing the communication or if, as explained above, he will always remain passive. The decision must be made separately for each commercial process, in which quality data are to be exchanged using QDX. Within the commercial process the roles of the two parties is constant and must not change. The decision is made in an individual partnership agreement (cf. Section 5.4). The role of the supplier is always active – in contrast to that of the customer. The supplier is therefore always responsible for transmitting his information to the customer.

The two types of communication (active/passive) are dealt with in detail in the following Sections (2.3.1 for active communication and 2.3.2 for passive communication).

As already mentioned in Section 2.1 large organisations often have so-called communication or routing systems which control communication between internal and external systems. In such cases the data are received by the routing system and then forwarded to the destination systems; the processing of the data in the destination system therefore does not take place at the instant the data are received. The fact that the receiving system acknowledges reception of the data does not indicate whether or not the data which have been sent have also arrived at the destination system and whether they could be processed there. For this reason a "processing" confirmation is introduced, with which the recipient confirms that the data have been received and processed correctly in the destination system. This guarantees the transparent transfer of data/information.

2.3.1 Active communication

The initiator of the commercial process initiates the communication and sends his information (data) to the other party. Once this information has been successfully processed in the destination system the second party sends a "processing" confirmation back to the initiator. This is followed by a response to the data which have been sent. The response is sent back to the initiator by the second party. Once the processing has been completed successfully, a "processing" confirmation is again sent back to the other party.

With "active" communication, both parties must send the generated data to the other party on their own responsibility. In this sense, both supplier and customer are responsible for the transmission of data.

2.3.1.1 The customer as initiator

The complaint process is a good means of explaining the communication between the two parties where the customer is the initiator of the commercial process.

The customer sends the data covering the complaint to the supplier. If the supplier's destination system has been able to process these data, it sends a "processing" confirmation back to the customer. This step is essential, so that the customer knows that the data he has sent have actually arrived in the destination system and have been processed successfully. With non-synchronous processing this step is even more important.

The receiving systems each acknowledge receipt of the data (complaint data and "processing" confirmation) [5].

[5] This usually covers the test/inspection reports.

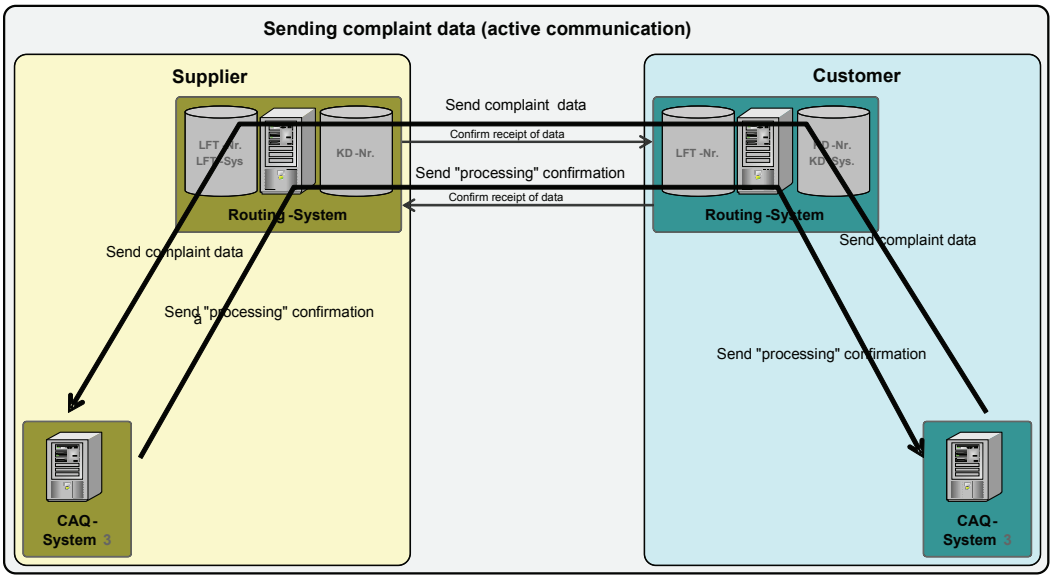

Fig. 2-8: Active communication based on the complaint process: sending complaint data

The response (in the form of an 8D report) is sent by the supplier to the customer. If the customer's destination system has received the 8D report and has been able to process it satisfactorily, the customer's system sends back a "processing" confirmation to the supplier. This step is essential, so that the customer knows that the data he has sent have actually arrived in the destination system and have been processed successfully. With non-synchronous processing this step is even more important.

Receipt of the data (8D report and "processing" confirmation) is acknowledged by the receiving systems. [5]

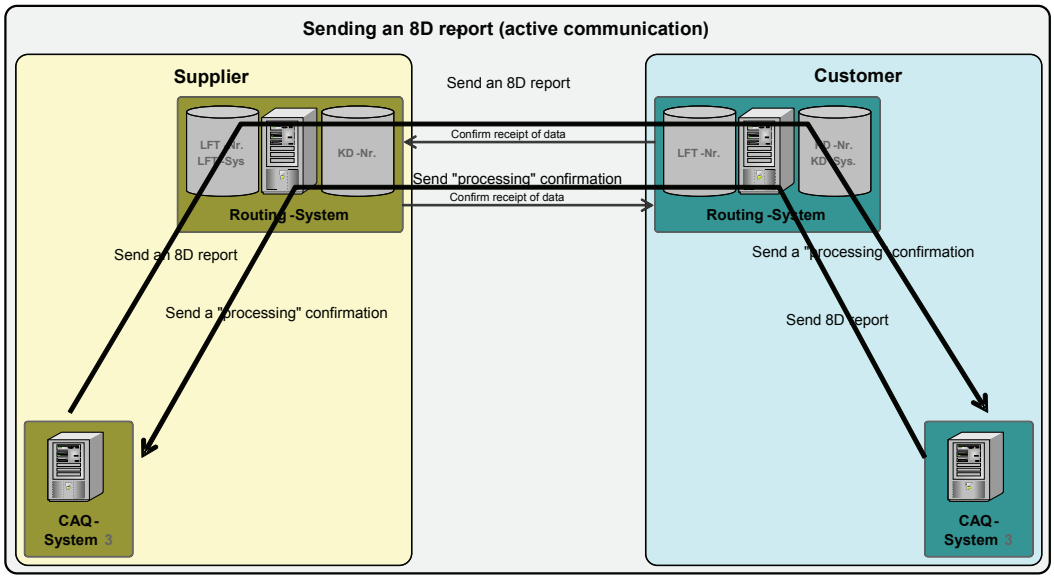

Fig. 2-9: Active communication based on the complaint process: sending an 8D report

2.3.1.2 The supplier as initiator

The initial sample submission process is a good means of explaining the communication between the two parties where the supplier is the initiator of the commercial process.

The supplier send the initial sample inspection report (PPF report) to the customer. If the customer's destination system has been able to process the PPF report satisfactorily, it sends a "processing" confirmation back to the supplier. This step is essential, so that the supplier knows that the data he has sent have actually arrived in the destination system and have been processed successfully. With non-synchronous processing this step is even more important).

The receiving systems acknowledge receipt of the data (PPF report and "processing" confirmation). [5]

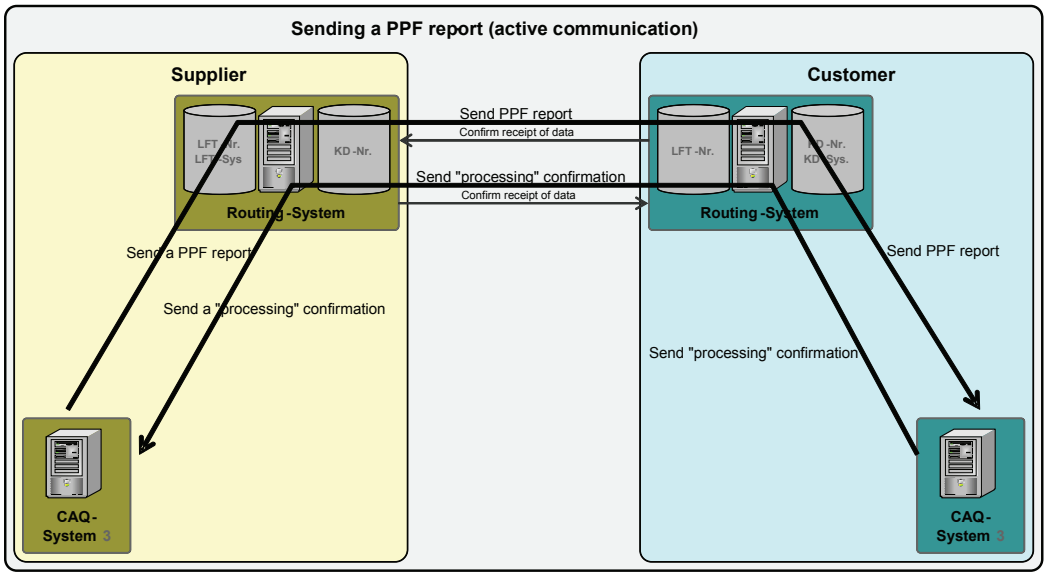

Fig. 2-10: Active communication based on an initial sample submission: sending a PPF report

The response (in the form of a PPF) is sent from the customer to the supplier. If the supplier's destination system has received the PPF and has been able to process it satisfactorily, the supplier's CAQ system sends a "processing" confirmation back to the customer. This step is essential, so that the customer knows that the data he has sent have actually arrived in the destination system and have been processed successfully. With non-synchronous processing this step is even more important).

Receipt of the data (PPF and "processing" confirmation) is acknowledged by the receiving systems. [5]

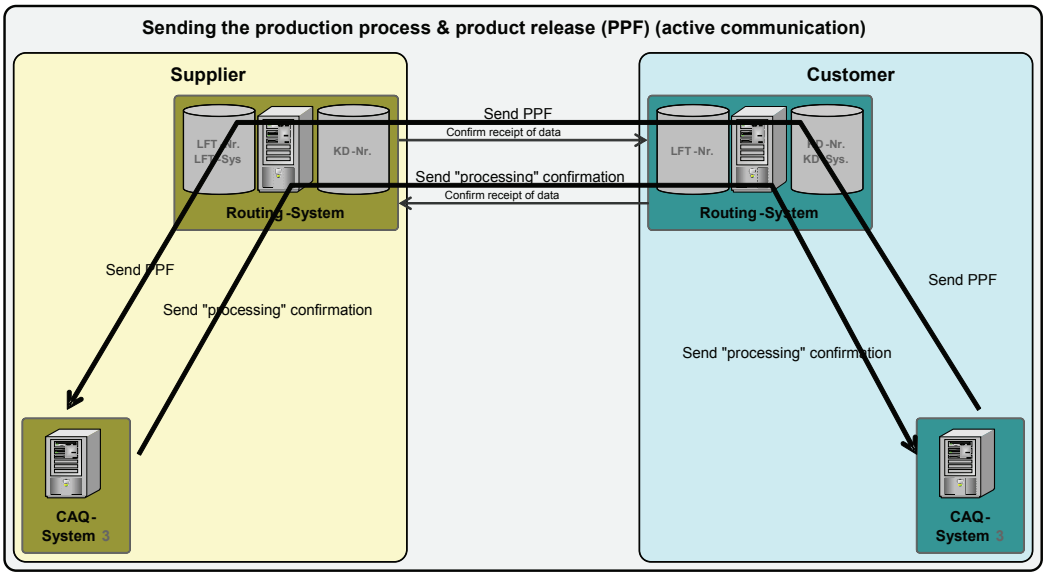

Fig. 2-11: Active communication based on an initial sample submission: sending the PPF

2.3.2 Passive communication

With passive communication the supplier must always initiate the communication with the customer. The customer, on the other hand, remains passive. If the supplier wishes to send data to the customer, the process is similar to that with active communication. The difference is that data from the customer are to be sent to the supplier and this is not carried out actively by the customer. Instead, it must be initiated by the supplier. The customer remains passive and merely makes the data available.

From the process initiator's stand-point this causes a difference when sending the initial message and also when sending the response.

With "passive" communication, therefore, it is always the supplier who is responsible for receiving customer data and for ensuring that his own data are sent to the customer. The supplier is therefore responsible both for the "bring" and "collect" processes.

Note : Passive communication is carried out typically by "polling" [6] *the other party. If a decision is made to use passive communication, the relevant systems must be able to operate with a basic number of continuous system accesses.*

2.3.2.1 The customer as initiator

F the customer is the initiator of the commercial process, the other party (the supplier) must ask the initiator if new information (data) is available and must then call up this information. Once the supplier's destination system has received the data made available by the initiator and processed them correctly, the supplier sends a "processing" confirmation to the initiator. This is followed by the response by the supplier, who sends his response back to the initiator. The supplier must then call up the processing confirmation for the response.

Communication between the two parties is explained as follows, taking the example of the complaint process.

The supplier must call up the complaint data from the customer. Once the supplier's destination system has processed the complaint data correctly, the supplier sends a "processing" confirmation back to the customer. The receiving system acknowledges receipt of the "processing" confirmation. [7]

[6] "Polling" describes the regular interrogation of thr server by a client, to determine whether new data/information is available.

[7] Reception of data which have been called up is not acknowledged within the framework of the technical protocol because this is usually already the technical response. Therefore, only the data which have been sent are acknowledged.

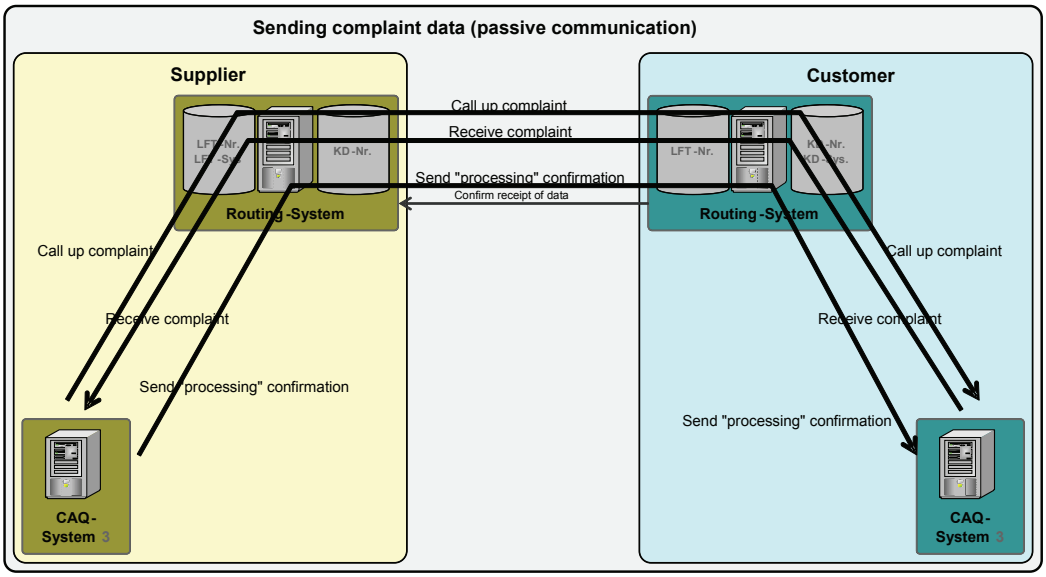

Fig. 2-12: Passive communication based on a complaint process: sending complaint data

The response (in the form of an 8D report) is sent by the supplier to the customer. The supplier must call up the "processing" confirmation from the customer to discover whether the 8D report has been processed correctly in the destination system. The receipt of the 8D report is acknowledged by the receiving system. [7]

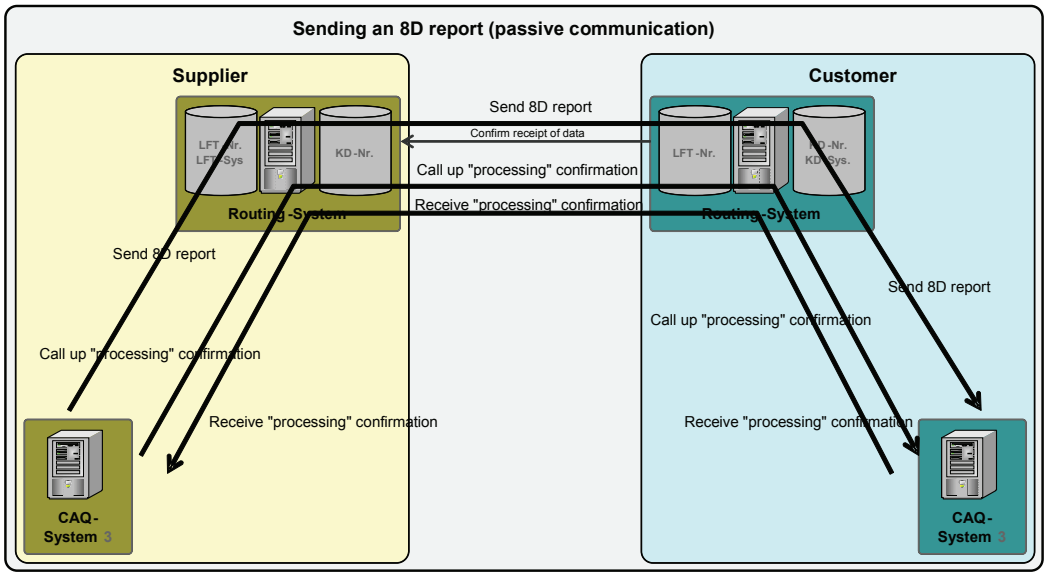

Fig. 2-13: Passive communication based on the complaint process: sending an 8D report

2.3.2.2 The supplier as initiator

If the supplier is the initiator of the commercial process, he must first send his information (Data) to the other party (the customer). However, the initiator must call up the "processing" confirmation. This is followed by a response by the customer. Again, the initiator must call up the response. Finally, the initiator sends the "processing" confirmation of the response to the customer.

Communication between the two parties is explained as follows, taking the example of the initial sample submission process.

The supplier sends the PPF report to the customer. The supplier must then call up the "processing" confirmation from the customer's destination system, to confirm that it has received the PPF report and processed it correctly. The customer's receiving system acknowledges receipt of the PPF report. [7]

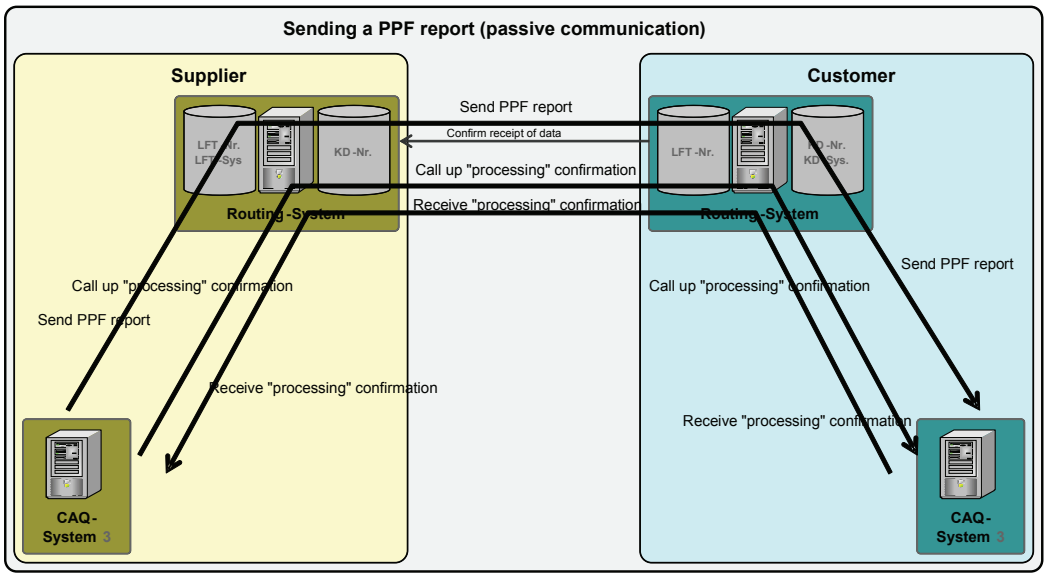

Fig. 2-14: Active communication based on an initial sample submission: sending a PPF report

The response (in the form of a PPF) must be called up by the supplier from the customer's system. Provided the supplier's system is able to process the PPF correctly, the supplier sends a "processing" confirmation back to the customer. The receipt of the "processing" confirmation is acknowledged by the customer's receiving system.[7]

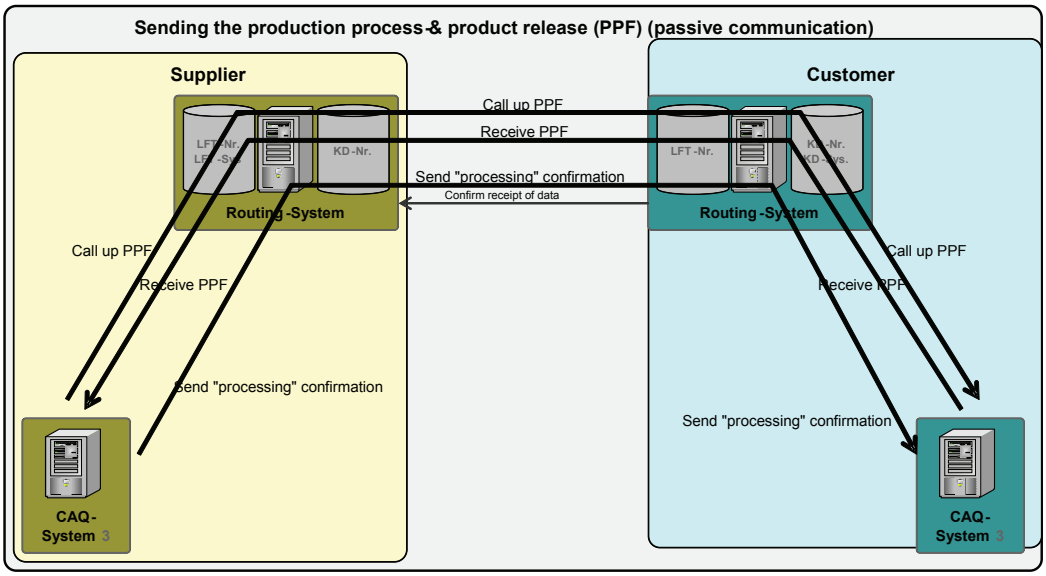

Fig. 2-15: Active communication based on an initial sample submission: sending the PPF

3 Communication procedure

The communication procedure must be laid down within the framework of an individual agreement between the parties. As part of such an agreement of this kind a definition must be made of the network (Internet, ISDN, ENX) which is to be used for communication.

If the customer has opted for "active communication" (see Section 2.3.1), OFTP must be used. In the case of "passive communication" (see Section 2.3.2) Webservice must be used.

The specifications set out in Section 4 must be met with both variants.

3.1 OFTP (active communication)

The **ODETTE-File-Transfer-Protocol** (OFTP) is a protocol for the electronic transfer of data between commercial parties and has been in use for over 20 years. It ensures the secure transport of messages between two parties, enabling the exchange of structured and also unstructured data. This exchange can be carried out as a point-to-point procedure or via added value network service providers, using ISDN, TCP/IP and also X.25. Communication using TCP/IP also includes data exchange via ENX and the Internet. However, the latter applies only to the use of OFTP Version 2.0.

The ODETTE-File-Transfer-Protocol guarantees :

Data independence : the OFTP can be used to transfer all kinds of user data.

Data protection : the exchange of a password is an essential component of the protocol. When a call comes in, the caller can be identified. This automatically bars unauthorised callers and prevents a connection. The ISDN and TCP/IP interface also provides the call-number check as a procedure to accept only incoming calls from known associates.

The OFTP 2.0 also provides the possibility of encoding, both at file level and at protocol level.

Data security : To protect the system against errors or loss of data, there is an integrated, automatic feedback from receiver to sender.

Compression : the OFTP enables data to be compressed and allows compressed data to be processed further. This performance characteristic reduces the number of characters to be transmitted and, therefore, the costs of transmission.

Restart after transmission errors : after a break in transmission caused by transmission errors, a fresh transmission can begin from the last data package which was correctly sent and received. The restart capability ensures that only the missing data need be transmitted – this reduces transmission time and also the costs of transmission.

Exchange of end-to-end-responses (EERPs) : OFTP supports the exchange of end-to-end responses, whereby the receiver confirms that the transfer of data is complete.

In addition to the above performance characteristics, OFTP 2.0 also guarantees :

Transaction security and confidentiality : with OFTP 2.0 transaction security and confidentiality are guaranteed both at line level and at file level. This ensures that sessions can be fully secured by encoding and authentication. Unauthorized third parties therefore have no access to the information in the transmission. For data transfer via the Internet, the entire session is secured by transport layer security (TLS)[8] which also provides authentication of the parties.

---

8 Transport Layer Security (TLS) is an encoding protocol for data transmission over the Internet. TLS 1.0 and 1.1 are standardised further developments of Secure Sockets Layer (SSL) 3.0.

When transmitting over other networks, the identities of the parties are checked at the beginning of an OFTP session via certificates. The files themselves can then be transmitted in both directions in encoded form.

Data integrity and authenticity : Data integrity and authenticity are guaranteed by OFTP 2.0 by means of digital signatures. These provide proof that the data are from the expected sender and have not been modified following transmission. For this purpose the sender "signs" the data with his "private key". The "public key" is used to check this signature. In order to check the signature each of the parties requires the other to provide the certificates with "public key".

Undeniability : OFTP 2.0 supports the exchange of end-to-end responses (EERP) and negative end-to-end responses (NERP), where the receiver confirms that the data transmission was complete or reports a defect in transmission. In this way the EERP also acts as evidence of reception. In addition the confirmation messages can contain digital signatures.

3.1.1 Transmitting QDX messages via OFTP

The transmission of QDX messages via operates basically to the same system for all message types. This system is shown in the following illustration.

No attempt is made here to cover the differences between OFTP V1 and V2 (particularly the security mechanisms of OFTP V2) because these are not relevant to a basic understanding of the exchange of QDX messages via OFTP.

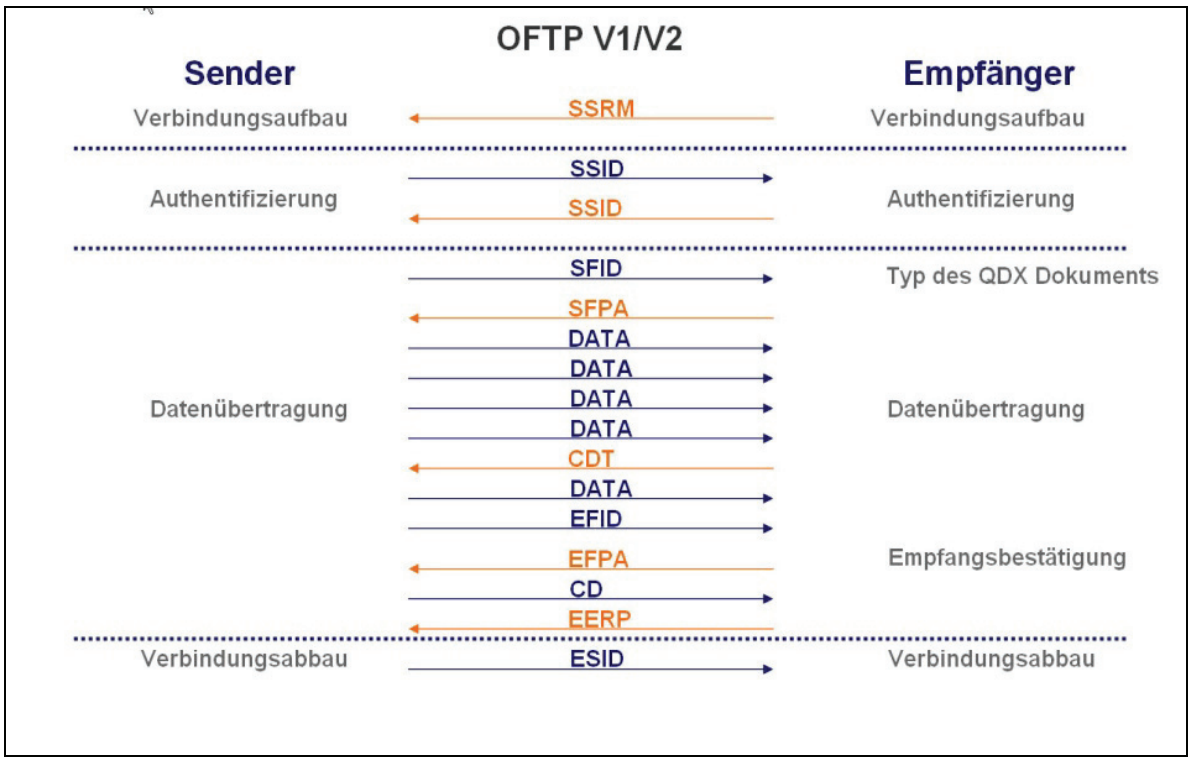

Fig. 3.1-1: Transmitting QDX messages via OFTP

The following comments assume that there is already a communication link via OFTP between the parties involved.

With a point-to-point connection the recipient is advised at protocol level , namely in the start file ID (SFID) of the type of QDX document which is to be transmitted. Based on this information a specific workflow can start on the receiving system , to process the message further.

In addition to transmitting QDX messages it will also be necessary in practice to send one or more attachments with a QDX message, as is explained in the following quality problem.

In the present example both variants of transmission of QDX data by OFTP are examined :

- transmitting QDX messages without attachments
- transmitting QDX messages with attachments

3.1.1.1 Transmitting QDX messages without attachments

Where QDX messages are transmitted without attachments the parties need only agree on the virtual file name for this type of data. This applies only to point-to-point connections. If the data are transmitted via a VAN, the operator frequently issues only one fixed virtual file name for all types of messages. In this case, the routing information required for further processing must be determined from the effective data.

3.1.1.2 Transmitting QDX messages with attachments

If attachments (*.jpg, *.wav,*.bmp) are to be sent with a QDX message, the parties must agree not only on the virtual file name but also whether the QDX message and attachments are to be transported in one or several transmissions (sessions). Transmission in several sessions requires greater organisational arrangements for the receiver and this is therefore excluded.

Transmitting a QDX message and attachments in a single session requires the individual files to be bundled together beforehand into a physical object (a data bundle). This is done, using a SOAP message in multipart MIME format. Using a SOAP envelope of this kind for bundling the data to be transferred is recommended particularly where further processing of the data is to be handled with a Webservice.

If the routing information has already been determined at protocol level, the data package can be transferred to the Webservice. Otherwise, the information required for routing must be taken from the SOAP header. In addition, references to the attachments are contained in the payload next to the actual QDX message. For further information see also Section 4.5.

The following illustration shows the contents of the SOAP message.

```
MIME-Version: 1.0
Content-type: multipart/mixed; boundary=mime-boundary; type=text/xml

--mime-boundary                                                    SOAP-Nachricht
Content-type: text/xml; charset=utf-8                              inkl. QDX-Dokument
Content-Transfer-Encoding: 8bit

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <env:Header>
        <wsa:To env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">urn:vda:qdx:empfänger</wsa:To>
        <wsa:From env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">
            <wsa:Address>urn:vda:qdx:sender</wsa:Address>
        </wsa:From>
        <wsa:Action env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">urn:vda:qdx:qdx-dokument</wsa:Action>
    </env:Header>
    <env:Body>
        <qer:QDXEnvelopeResponse xmlns:qer="www.vda-qmc.de/qdx/QDXEnvelopeResponse_V2.0.xsd">
            <qer:Code>201</qer:Code>
            <qer:CodeDescription>Request succeeded</qer:CodeDescription>
            <qer:CodeDetails>Anfrage erfolgreich durchgeführt</qer:CodeDetails>
            <qdx:QDXComplaint xmlns:qdx="www.vda-qmc.de/qdx/QDXComplaint_V2.0.xsd">
                ...
            </qdx:QDXComplaint>
        </qer:QDXEnvelopeResponse>
    </env:Body>
</env:Envelope>

--mime-boundary                                                    Attachment 1
Content-type: image/jpg
Content-ID: 1
Content-Transfer-Encoding: binary

...binäres jpg-Bild....

--mime-boundary                                                    Attachment 2
Content-type: video/mpeg
Content-ID: 2
Content-Transfer-Encoding: binary

...binäres mpeg-Video...
```

Fig. 3.1-2: Structure of a SOAP message made up of a QDX complaint with two attachments

3.1.2 Transmitting complaints via OFTP

Following the general notes earlier in this publication regarding the transmission of QDX messages, this present section examines the process of transmitting a complaint.

The process is initiated by the customer, who sends the complaint as a QDXComplaint to the supplier. The supplier confirms that the complaint has been processed correctly in his CAQ system by sending back a confirmation via QDXAcknowledgeComplaint.

If the supplier needs the customer to repeat the transmission of the complaint message, he can request a repeat transmission by sending an appropriate message (QDXResetDeliveryStatus Complaint).

A separate virtual file name must be agreed for all three QDX message types.

The following picture illustrates the process.

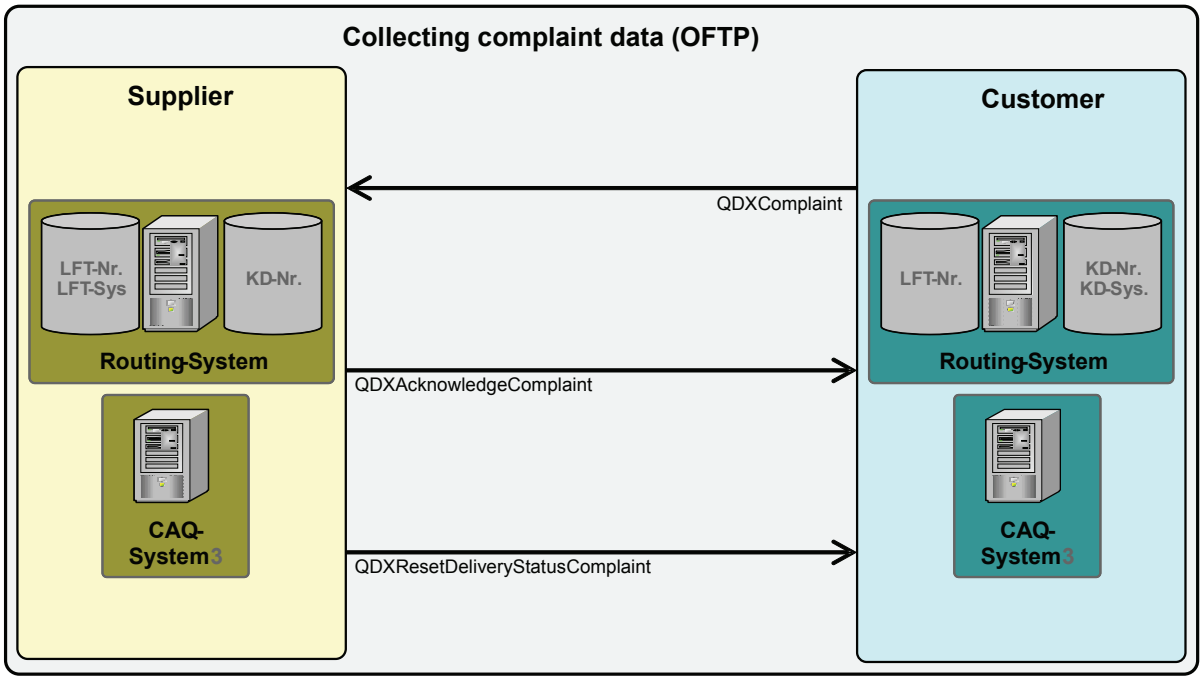

Fig. 3.1-3: Transmitting a complaint message via OFTP

3.1.3 Transmitting 8D reports via OFTP

As a response to a complaint which is accepted, the supplier sends an 8D report (QDXReport8D) to the customer. The customer confirms the correct processing in his CAQ system of the 8D report which he has received, by sending back an acknowledgement (QDXAcknowledgeReport8D) to the supplier.

A separate virtual file name must be agreed for the two QDX message types.

The following picture illustrates the process.

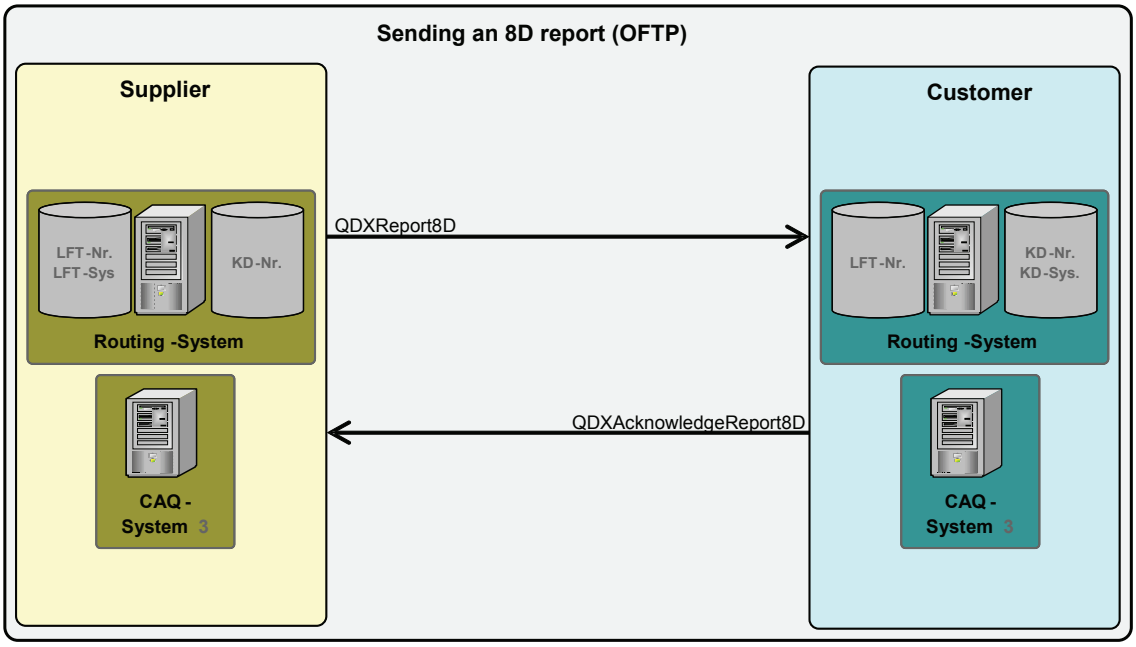

Fig. 3.1-4: Sending an 8D report via OFTP

3.2 Webservices (passive communication)

The procedure for communication via Webservice is the same as that described in Section 2.3.2, where the structure of the communication is always established by the supplier. When initiating the communication, the supplier must know the methods to be called up and therefore this present section also defines the Webservice methods, including the call-up parameters.

For this reason this section cannot formulate a method which applies in all cases and, therefore, a description is provided here only of the Webservice methods for the complaint process.

Waiting times[9] and access intervals to the Webservice must be defined in an individual agreement between the parties. In addition, the time-out time must be specified, after which a client breaks off the enquiry[10]. The customer must answer the request within the time-out time and send back the data required (e.g., a QDXComplaint including all attachments).

Note : If complaint data are generated only when the Webservice method is called up, particular care must be taken to comply with the specified time-out time.

3.2.1 Authentication

For authentication with regard to the Webservice the basic-authentication[11] procedure is used. Here, the user name and password are transmitted in coded form. HTTPS is recommended as the transmission protocol, where an encoded connection is established, so that user name and password are protected against unauthorized access from outside.

3.2.2 Fetching complaints

In the complaint process the initial message – the complaint (QDXComplaint) – is generated by the customer. Because of the passive communication, the communication structure must be established by the supplier, who must call up the message.

To optimize communication the sequence described in Section 2.3.2 is expanded by two enquiries :
To call up complaints the supplier must first call up a list of all the complaints which he is to collect. The supplier can cause a complaint to be sent again.

---

[9] For example, waiting times between sending an8D report and calling up the "processing" confirmation regarding the 8D report which has been sent)
[10] Standard time-out time : 120 s.
[11] For more information on basic authentication: http://tools.ietf.org/html/rfc2617

The following illustration gives an overview of the Webservice methods and their logical sequences.

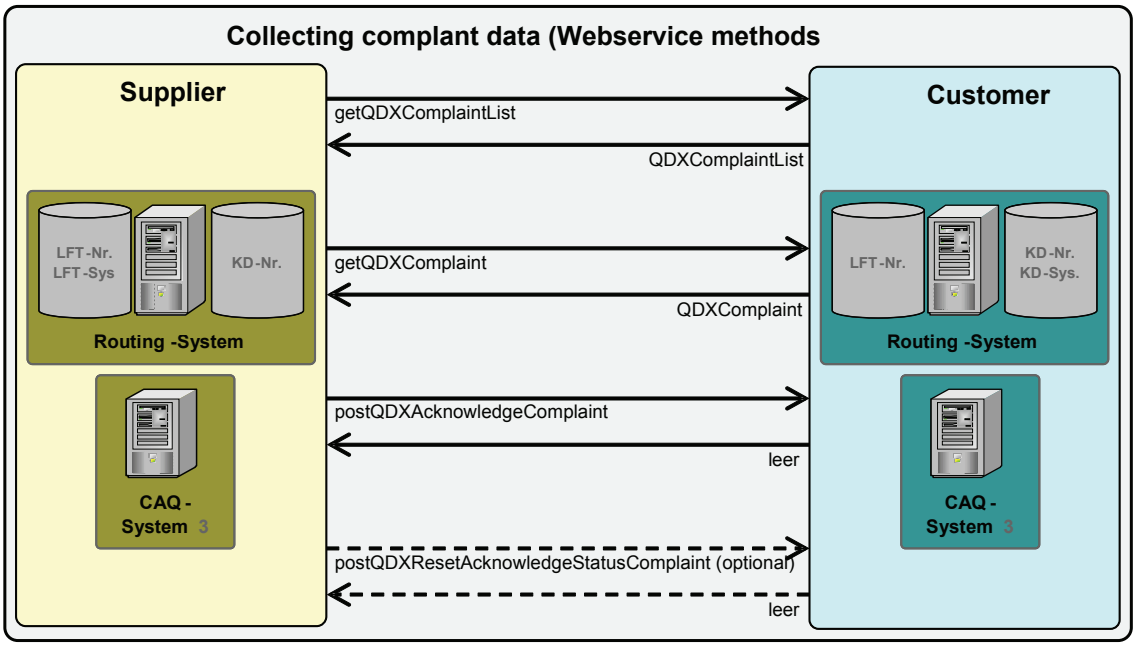

Fig. 3-5: Webservice methods for calling up complaints

getQDXComplaintList: With this method the supplier (the CAQ system) must call up a list of all the complaints which he must collect. The list contains the IDs of the complaints issued by the customer. Based on the ID the supplier can call up the method described below.

getQDXComplaint: The supplier uses this to call up the complaints individually, based on the IDs he has received.

postQDXAcknowledgeComplaint: After the QDXComplaint has been successfully processed in the supplier's CAQ system, the supplier must confirm this. This step is obligatory and must always be carried out after the successful processing of a QDXComplaint! Once confirmed, a QDXComplaint will not be repeated with the next call-up of “getQDXComplaintList”.

postQDXResetAcknowledgeStatusComplaint: If there is a reason why a QDXComplaint which has been called up and acknowledged must be called up again by the supplier, the status

of the QDXComplaint can be reset with this method.[12]. This step can be carried out if required and optionally by the supplier.

Each Webservice method requires information from the system making a call-up in order to provide the correct data. These transfer parameters are issued as a QDX document in "Swap-Body" as an HTTP request. The result of the method call-up is transmitted as a QDX document within the framework of the HTTP response.

The following table shows the transfer parameter, result and possible status codes for each Webservice method. The QDX documents required for the transfer parameters and for the result are described in detail in Section 4.4. The status codes are described in Section 5.3.

Method	Transfer parameter	Result	Status codes
getQDXComplaintList	**QDXComplaint ListRequest**	**QDXCom plaintList**	**200, 400**
getQDXComplaint	**QDXComplaint Request**	**QDXCom plaint**	**201, 401, 402, 403**
postQDXAcknowledge Complaint	**QDXAcknowledge Complaint**	**leer**	**202, 401, 402, 403, , 404, 405, 406**
postQDXResetAckno wledgeStatusComplai nt	**QDXResetAcknow ledgeStatusCompl aint**	**leer**	**203, 401, 402, 403**

Table 3-1: Description of Webservice methods for collecting complaints

[12] A reset of the status of a QDXComplaint has no automatic (!) effect on the monitoring of timing. as a rule the timing monitoring is not adjusted because here there is a tendency for the supplier to be responsible.

3.2.3 Sending an 8D report

In the standard complaint process the 8D report represents the factual response and is drawn up by the supplier. Because the communication here is established by the party creating the document (the supplier), the communication is more simple than calling up complaints.

The following illustration gives an overview of the Webservice methods and their logical sequence for the 8D report. Further response types (such as QDXShortConfirmation or QDX FieldFailureResponse) can be used in the same way.

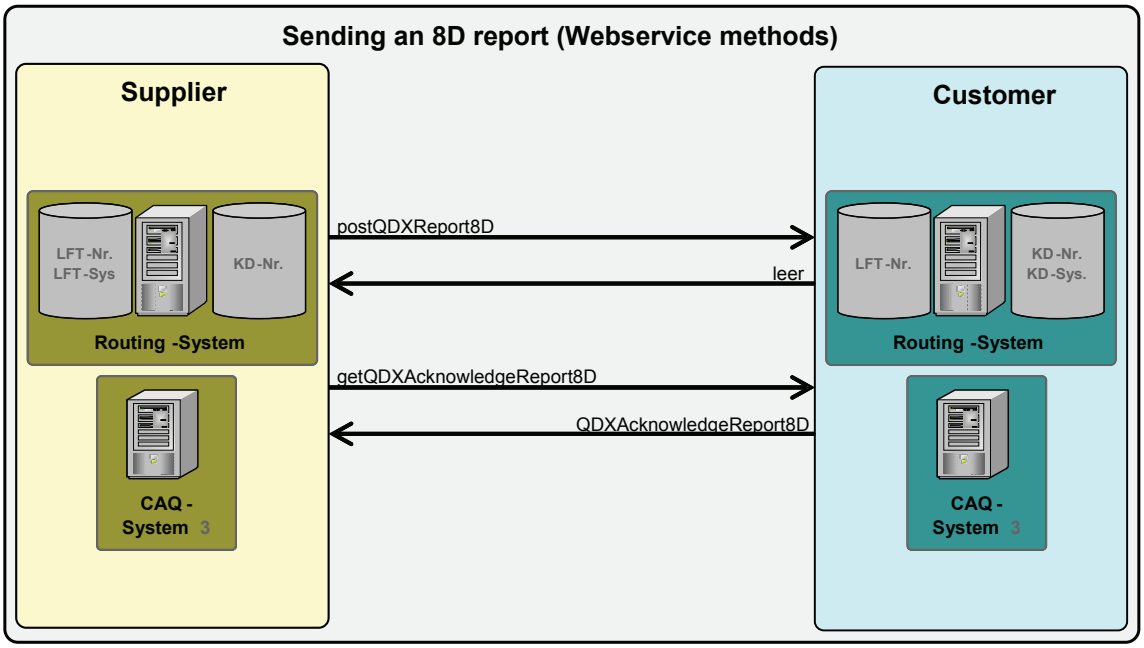

Fig. 3-6: Webservice methods for handling 8D reports

postQDXReport8D: The supplier can use this to set up an 8D report regarding a complaint in the customer's CAQ system.

getQDXAcknowledgeReport8D: To ensure that the customer's CAQ system was able to process the 8D report correctly, the supplier must interrogate this with this method. This step is obligatory and must always be carried out after sending a QDXReport8D! (the waiting time between executing the two methods must be defined in an individual agreement between the parties - see Section 5.4).

Each Webservice method requires information from the system making the call-up in order to provide the correct data. These transfer parameters are issued as a QDX document in "Swap-Body" as an HTTP request. The result of the method call-up is transmitted as a QDX document within the framework of the HTTP response.

The following table shows the transfer parameter, result and possible status codes for each Webservice method. The QDX documents required for the transfer parameters and for the result are described in detail in Section 4.4. The status codes are described in Section 5.3.

Method	Transfer parameter	Result	Status code
postQDXReport8D	**QDXReport8D**	**empty**	**204, 401, 402, 403**
getQDXAcknowledgeReport8D	**QDXAcknowledgeReport8DRequest**	**QDXAcknowledgeReport8D**	**205, 401, 402, 403**

Table 3-2: Description of the Webservice methods for sending the factual response

4 The data package format

In the system landscapes of today's large-scale organisations, communication with external systems is often controlled by a central routing system. Therefore no direct communication takes place between the two end-systems (CAQ systems) - instead, there are other systems between them. In order to direct QDX documents to the correct CAQ system, the routing systems need to be able to "look into" the QDX documents to find defined information for routing purposes. To prevent the situation where each intermediate system must "understand" QDX, it has been laid down that the QDX documents should be transmitted within a SOAP message[13] – no matter what transmission technology is used.

The "meta" information required for processing the message and for routing it in intermediate systems is set out in the header of the SOAP message (also referred to as the SOAP envelope) – for a more precise description, see Section 4.1. The actual QDX document is in the SOAP-Body (cf. Section 4.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
        <env:Header>
        </env:Header>
        <env:Body>
        </env:Body>
</env:Envelope>
```

Fig. 4-1: Structure of the SOAP envelope

SOAP has been selected as the transport structure for the QDX message because SOAP is an open standard of the W3C[14] and may be regarded already as a standard for Webservices within the framework of data transfer. Furthermore, SOAP can be used independently of the transfer protocol which is used.

[13] See also the definition of the SOAP version 1.2 used in http://www.w3.org/TR/soap12/

[14] W3C = World Wide Web Consortium

The SOAP message itself is sent in MIME format[15] as a multi-part message[16] - in other words, the message is split into several parts. The first part of the message contains the SOAP message with the QDX documents. The optional following parts of the multi-part message contains possible attachments (see Section 4.5). The following illustration shows the make-up and structure of the data transmitted :

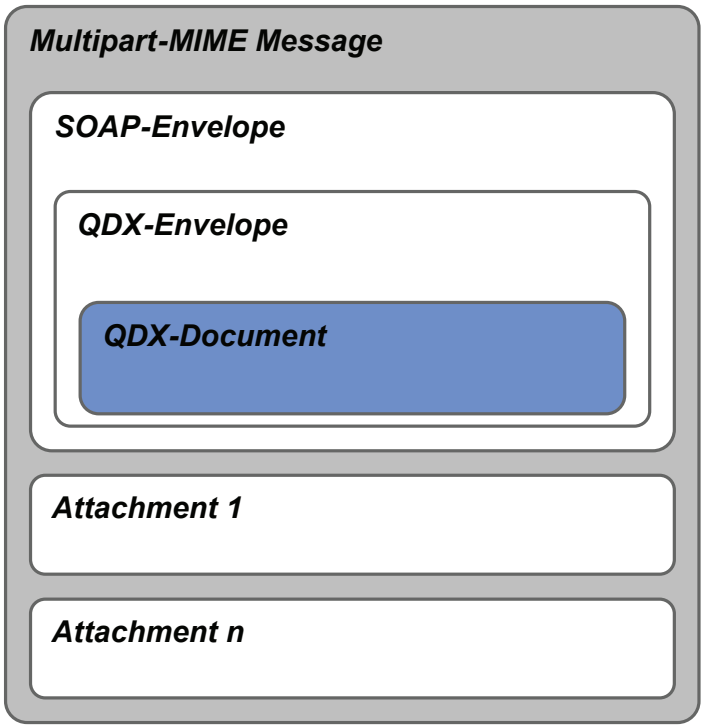

Fig. 4-2: Structure of the data transmitted

An example of a multi-part MIME message is shown in Fig. 4.3.

[15] See also the definitions under : http://tools.ietf.org/html/rfc2045
[16] See also the definition under: http://tools.ietf.org/html/rfc2387

```
MIME-Version: 1.0
Content-type: multipart/mixed; boundary=mime-boundary; type=text/xml

--mime-boundary
Content-type: text/xml; charset=utf-8
Content-Transfer-Encoding: 8bit

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:wsa="http://www.w3.org/2005/08/addressing">
            ...
</env:Envelope>

--mime-boundary
Content-type: image/jpg; name="bild.jpg"
Content-Transfer-Encoding: binary

...binäres jpg-Bild....

--mime-boundary
Content-type: video/mpeg; name="video.mpeg"
Content-Transfer-Encoding: binary

...binäres mpeg-Video...
```

Fig. 4-3: Example of a multi-part MIME message

4.1 Header

The information transmitted in the SOAP header is essential for further processing in the intermediate systems (routing systems). Most importantly, this information includes the identifications of the receiver and the sender, as well as the type of QDX document. This last item is useful for the CAQ systems which, from reading the SOAP envelope, will know what type of document is to be processed.

The W3C standard "WS-Addressing“[17] has been specified as the system for transmitting "meta" information. Like SOAP, the WS-Addressing standard is an open standard and is widely used – particularly when implementing Webservices.

The data in the fields of the WS-Addressing standard must be shown in the form of a URI[18]. The system to be used for this has been specified as URN[19]. An application for the use of the name space "vda" has been lodged with the relevant name space administration organisation. The VDA[20] will be able to log its own name spaces[21] such as "qdx".

[17] WS-Addressing = Webservice addressing. See also the definition under: http://www.w3.org/2002/ws/addr/
[18] URI = Uniform Resource Identifier
[19] URN = Uniform Resource Name
[20] VDA = Verband der Automobilindustrie e.V.
[21] iana.org

The data in the SOAP header must therefore be shown as follows: "urn:vda:qdx:*daten*".

This results in the following example of a SOAP header :

```
<env:Header>
        <wsa:To env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">
                urn:vda:qdx:empfänger
        </wsa:To>
        <wsa:From env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">
                <wsa:Address>
                        urn:vda:qdx:sender
                </wsa:Address>
        </wsa:From>
        <wsa:Action env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">
                urn:vda:qdx:qdx-dokument
        </wsa:Action>
</env:Header>
```

Fig. 4-4: Structure of a SOAP header

Receiver : The identifier number of the other party must be entered as the receiver. If the message is sent by the customer, the supplier number from the customer's records is entered. If the supplier is the sender, he must enter the customer number which the customer has allotted to him.
Example : 12345678A

Sender: The sender must enter his own identifier number. If the message is sent by the customer, his customer number from his own numbering system is entered here. If the supplier is the sender, he must enter the supplier which the customer has allotted to him.
Example : 1234567800

QDX-Document: The reference for the QDX document is analogous to the reference of the system files.
Example : QDXReport8D

The identifier of the receiver and the sender can also be extended with a system reference. This can be a system number which is known in-house, or any other figure which can be used for the purpose of forwarding QDX documents to the correct CAQ system. The system identifier is separated from the receiver / sender by a full-stop :

System identifier : urn:vda:qdx:sender.systemkennung or urn:vda:qdx:empfänger.systemkennung
Example : urn:vda:qdx:12345678A.sys-id123

Entering a system identifier is optional. However, if a response to a process by the other party is to be sent with the SOAP envelope, the system identifier should be stated which was transmitted with the initial message from the other party (see also Section 2.1). If no system identifier was included, this can also be left blank in the response.

4.2 Body

The SOAP body contains the QDX document. However, the QDX document is also enclosed in a further package – the QDX envelope. The contents required in the QDX envelope depends on the associated integration process (active or passive communication - see Section 2). The QDX envelope is described in more detail in Section 4.3. The associated WSDL is set out in Section 5.1.

4.3 QDX envelope

When transmitting a QDX document it is usually necessary to send further information (for authentication purposes, for example). Because of this, the QDX document and the additional information are wrapped in their own envelope – the QDX envelope. The contents and, therefore, the make-up of the QDX envelope depends initially on the integration process (see Section 2). With active communication a QDX envelope is sufficient. With passive communication a method is called up (generally in a request), whose response delivers the necessary information. Because of this, two QDX envelopes have been defined for passive communication.
The definitions of the associated XSDs are set out in Section 5.2.

4.3.1 QDXEnvelope (active communication)

For active communication, the same QDX envelope is used in terms of elements as for the http request with passive communication (see Section 4.3.2). To prevent any confusion

which the term "request" might cause with OFTP, a separate QDX envelope has been defined, with the title QDXEnvelope.

```
<env:Body>
        <qer:QDXEnvelope xmlns:qer="www.vda-qmc.de/qdx/QDXEnvelope_V2.0.xsd">
                <qdx:QDXReport8D xmlns:qdx="www.vda-qmc.de/qdx/QDXReport8D_V2.0.xsd">
                        ...
                </qdx:QDXReport8D>
        </qer:QDXEnvelope>
</env:Body>
```

Fig. 4-5: Make-up of a QDXEnvelope

4.3.2 QDXEnvelopeRequest (passive communication)

The QDX document is also packed in a QDX envelope. Apart from the document, which is needed for the http request, the envelope contains no further information.

```
<env:Body>
        <qer:QDXEnvelopeRequest xmlns:qer="www.vda-qmc.de/qdx/QDXEnvelopeRequest_V2.0.xsd">
                <qdx:QDXReport8D xmlns:qdx="www.vda-qmc.de/qdx/QDXReport8D_V2.0.xsd">
                        ...
                </qdx:QDXReport8D>
        </qer:QDXEnvelopeRequest>
</env:Body>
```

Fig. 4-6: Make-up of a QDXEnvelopeRequest

4.3.3 QDXEnvelopeResponse (passive communication)

With the response to a request, the QDX document must be accompanied by (or replaced by) status information for the request to be processed. For this reason, this QDX envelope contains a status code, which can be processed by machine, a standard description for the status code and a special description for the status code, which may if appropriate, contain a description of the cause of the defect.

As the actual response to the original request, the QDX document is sent as well, if the request was successful. If it was not, the

QDX envelope contains only the status code with the description of the defect.

```
<env:Body>
        <qer:QDXEnvelopeResponse xmlns:qer="www.vda-qmc.de/qdx/QDXEnvelopeResponse_V2.0.xsd">
                <qer:Code>201</qer:Code>
                <qer:CodeDescription>Request succeeded</qer:CodeDescription>
                <qer:CodeDetails>Anfrage erfolgreich durchgeführt</qer:CodeDetails>
                <qdx:QDXComplaint xmlns:qdx="www.vda-qmc.de/qdx/QDXComplaint_V2.0.xsd">
                        ...
                </qdx:QDXComplaint>
        </qer:QDXEnvelopeResponse>
</env:Body>
```

Fig. 4-7: Make-up of a QDXEnvelopeResponse

All status codes are described in Section 5.2.10.

4.4 New QDX documents

To support the integration process (communication process) further methods have been defined which (among other functions) ensure that the QDX document has arrived in the destination system. In addition, in the case of a passive communication, this information and further information must be checked initially, before QDX documents can be sent and processed. The QDX documents required for this are described in this section. The documents are then broken down into parts, depending on whether they already contain the relevant information or are required merely in order to interrogate the information.

The QDX documents which transport factual information are documents which already exist : QDXComplaint, QDXReport8D, QDXShortConfirmation and QDXFieldFailure-Response. The QDX documents described here refer only to the use of QDXComplaint and QDXReport8D. If further documents are introduced in the complaint process, a check must first be made to see whether the documents are sent from the customer of the supplier.

If the document newly introduced for use is sent by the customer, the QDX documents associated with QDXComplaint (this can be seen from their reference) must be adapted and aligned to comply with the identification of the new QDX document.

If the document newly introduced for use is sent by the supplier, the same procedure applies as for the documents which belong to the QDXReport8D. If the document is used by both parties, all the documents described must be adapted and aligned.

The definition of system files for the new QDX documents is set out in Section 5.2.

4.4.1 QDX documents with technical information

This section describes QDX documents which transport information covering technical processing to the receiving CAQ system.

4.4.1.1 QDXAcknowledgeComplaint

This document is used to confirm receipt and the correct processing of a QDXComplaint in the destination system. In this way, the sender of the complaint (the customer) is certain that the QDX document has arrived with the supplier and that work can begin on eliminating the defect.

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The first step is to indicate the customer (and customer plant), to whom receipt of the complaint must be acknowledged. The customer number is the number which the customer has allocated to himself.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** Enter the number which the customer has given to the QDXComplaint. This is usually the number of the test/inspection report or the complaint number.
Complaint / ComplaintItemID	Obligatory	**Complaint number** The complaint whose processing is to be acknowledged must be entered here. This is the number issued by the customer.
Complaint / RevisionID	Optional	**Version number** The version number can also be entered as an option to ensure the unambiguous identification of the complaint.
Complaint/RevisionDateTime	Obligatory	**Version date** To ensure the unambiguous identification of the complaint, the version date issued by the customer for the complaint must be entered in addition to the DocumentID.

Table 4-1: Make-up of a QDXAcknowledgeComplaint

4.4.1.2 QDXResetAcknowledgeStatusComplaint

In some circumstances it may be necessary to call up a QDXComplaint again, after it has been processed. This document can be used to reset the processing status and/or transmission status. Depending on the type of communication (active or passive) the QDXComplaint will be sent again by the source system or it can be called up again by the destination system.

No version number or version date is included in the details of the document because it is always the current QDXComplaint which is sent.

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The first step is to indicate the customer (and customer plant) from which the complaint whose processing status is to be reset has come. The customer number is the number which the customer himself has issued.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** Enter the number which the customer has given to the QDXComplaint. This is usually the number of the test/inspection report or the complaint number.
Complaint/ComplaintItemID	Obligatory	**Complaint number** Enter here the complaint whose processing status is to be reset. The number issued by the customer must be used here.

Table 4-2: Make-up of a QDXResetAcknowledgeStatusComplaint

4.4.1.3 QDXAcknowledgeReport8D

Similar to the confirmation of correct processing by the supplier of a QDXComplaint, the customer must also confirm the processing of a QDXReport8D to the customer. This assures the supplier that his 8D report has arrived in the customer's destination system and has been correctly processed there. In certain circumstances this may be relevant for compliance with timing deadlines and, therefore, the supplier assessment.

There is no provision for resetting the processing status. If the 8D report is required again, this can be requested in the form of an up-dated QDXComplaint.

Data field	Obligatory/ optional	Description
SellerParty / ID	Obligatory	**Supplier number** The supplier, to whom processing of the 8D report must be confirmed, is entered here. The customer must use the number which he has issued for the supplier.
SellerParty / AdditionalID	Optional	**Additional supplier number** In addition to the supplier number a further number can be entered, to identify the supplier – for example the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** The number of the QDXComplaint sent by the customer must be entered here. The number issued by the customer must be used. This is usually the number of the test/inspection report or the complaint number.
Complaint / ComplaintItemID	Obligatory	**Complaint number** The number of the complaint, to which the 8D report refers , must be entered here. The customer must use his own reference number.
Report8D / DocumentID	Obligatory	**8D report number** The 8D report, whose processing is to be confirmed, must be entered. This is the number issued by the supplier.

Data field	Obligatory/ optional	Description
Report8D / RevisionID	Optional	**Version number** The version number of the 8D report can be entered here to ensure unambiguous identification.
Report8D/RevisionDateTime	Obligatory	**Version date** The version date of the 8D report must be entered in addition to the document ID, to ensure unambiguous identification.

Table 4-3: Make-up of a QDXAcknowledgeReport8D

4.4.1.4 QDXComplaintList

This document is the only exception among the QDX documents with technical information, because it is required exclusively for passive communication. With active communication the QDXComplaint is sent directly to the supplier; with passive communication the supplier must call up the QDXComplaint. To ensure that the supplier's system knows how many and what complaints are to be called up, the supplier must first call up a list containing the complaints to be collected. With this list the supplier will know how many QDXComplaints (and which) are to be called up.

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The first step is to indicate the customer (and customer plant) from which the complaints can be collected. The customer number is the number which the customer has issued for himself.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** The number of the QDXComplaint sent by the customer must be entered here. This is usually the test/inspection report number or the complaint number.
Complaint/ComplaintItemID	Obligatory (1..n)	**Complaint number** This is the number of the complaint which the supplier can call up. The number issued by the customer must be used here. It is not necessary to enter the version number or date, as it is always the latest QDXComplaint which is available. The data field can be used as often as desired and must be used at least once.

Table 4-4: Make-up of a QDXComplaintList

4.4.2 QDX documents for requesting information

Within the framework of passive communication it is essential for the supplier to call up customer information (QDX documents). For this purpose parameters must be provided, so that the customer's processing system can return the correct QDX document with the correct information.

4.4.2.1 QDXComplaintListRequest

This document is used to interrogate the list of complaints to be collected (see Section 4.4.1.4).

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The supplier must state from which customer (and customer plant) the list of complaints to be collected is to be interrogated. The customer number is the number issued by the customer for himself.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.

Table 4-5: Make-up of a QDXComplaintListRequest

4.4.2.2 QDXComplaintRequest

This document must be used to call up a QDXComplaint.

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The supplier must enter the customer (and customer plant) from which the complaints are to be called up. The customer number is the number which the customer has issued for himself.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** The number of the QDXComplaint sent by the customer must be entered here. Use the number issued by the customer. This is usually the number of the test/ inspection report or the complaint number.
Complaint/ComplaintItemID	Obligatory	**Complaint number** The number of the complaint which is to be called up. Use the number issued by the customer. It is not necessary to enter the version or version date, because it is always the latest QDXComplaint which is available.

Table 4-6: Make-up of a QDXComplaintRequest

4.4.2.3 QDXAcknowledgeReport8DRequest

To receive an acknowledgement that an 8D report which has been sent has been received in the destination system and has been processed correctly, this document must be used as the interrogation.

Data field	Obligatory/ optional	Description
BuyerParty / ID	Obligatory	**Customer number** The first step is to enter the customer (and customer plant) which is to be asked whether an 8D report has been processed correctly. The customer number is the number which the customer has issued for himself.
BuyerParty / AdditionalID	Optional	**Additional customer number** A further number can be given in addition to the customer number as a customer identification – for example, the DUNS number.
Complaint / DocumentID	Obligatory	**Document number** Enter the number of the QDXComplaint which was sent by the customer. The number issued by the customer must be used. This is usually the number of the test/inspection report or the complaint number.
Complaint/ComplaintItemID	Obligatory	**Complaint number** Enter the complaint covered by the 8D report, whose processing status is to be checked. The supplier must enter the customer's number here.
Report8D / DocumentID	Obligatory	**8D report number** The 8D report, whose processing status is to be confirmed must be entered here. This is the number issued by the supplier.

Data field	Obligatory/ optional	Description
Report8D / RevisionID	Optional	**Version number** The version number of the 8D report can be entered to ensure unambiguous identification.
Report8D /RevisionDateTime	Obligatory	**Version date** The version date of the 8D report must be entered in addition to the DocumentID to ensure unambiguous identification.

Table 4-7: Make-up of a QDXAcknowledgeReport8Drequest

4.5 Attachments

As a rule, attachments (sometimes large items) are sent with a QDX document. All attachments to a QDX document are contained within the data package in the MIME message. In this case, the expression "SOAP with attachments“ is also used – that is, a SOAP message with any number of attachments.

It may be sensible to restrict the size of attachments – this can be arranged by agreement between the two parties (cf. Section 5.4).

The referencing of attachments is not carried out on the basis of the file name but on the basis of an unambiguous ID contained within the MIME message. This enables two problems to be avoided :

- Identical file name for different attachments.
- Use of characters not meeting ASCII requirements in naming the file.

```
--mime-boundary
Content-type: image/jpg
Content-ID: 1
Content-Transfer-Encoding: binary

...binäres jpg-Bild....
```

[22]

Fig. 4-8: Indicating attachments

[22] Base64 must be used as the coding procedure. In this connection see the definitions in: http://tools.ietf.org/html/rfc4648

Because of this, the way the reference is indicated in the QDX documents has also been modified : the name of the data field "URL“ has been changed to "AttachmentID“ and this must be mapped in the Content-ID.

```
<MimeReference>
        <AttachmentID>1</AttachmendID>
        <RevisionDateTime>2007-11-02T11:58:01.499+01:00</RevisionDateTime>
        <Filename>Bild1.jpg</Filename>
</MimeReference>
```

Fig. 4-9: MIME reference

The "wholesale" principle applies when sending a QDX message – all the information is sent, every time. This principle also applies to the attachments – all current attachments are transmitted every time. If an attachment is no longer referenced and sent in a new version of a QDX message (e.g., QDXComplaint) the link between the attachment and the current version of the QDX document (a QDXComplaint in this example) is also dissolved with the other party (the supplier in this example). The supplier is also entitled to delete the attachment completely. However, this is not recommended, in the interests of maintaining comprehensive documentation/archiving.

4.6 Summary

This results in the following illustration of a SOAP message with a QDX document in multi-part MIME format, including attachments. An example of a complete QDX document is provided in Section 5.6. The individual elements in the MIME message are separated graphically by a separating line.

```
MIME-Version: 1.0
Content-type: multipart/mixed; boundary=mime-boundary; type=text/xml

--mime-boundary                                                    SOAP-Nachricht
Content-type: text/xml; charset=utf-8                              inkl. QDX-Dokument
Content-Transfer-Encoding: 8bit

<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <env:Header>
        <wsa:To env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">urn:vda:qdx:empfänger</wsa:To>
        <wsa:From env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">
            <wsa:Address>urn:vda:qdx:sender</wsa:Address>
        </wsa:From>
        <wsa:Action env:role="http://www.w3.org/2003/05/soap-envelope/role/next" env:relay="true">urn:vda:qdx:qdx-dokument</wsa:Action>
    </env:Header>
    <env:Body>
        <qer:QDXEnvelopeResponse xmlns:qer="www.vda-qmc.de/qdx/QDXEnvelopeResponse_V2.0.xsd">
            <qer:Code>201</qer:Code>
            <qer:CodeDescription>Request succeeded</qer:CodeDescription>
            <qer:CodeDetails>Anfrage erfolgreich durchgeführt</qer:CodeDetails>
            <qdx:QDXComplaint xmlns:qdx="www.vda-qmc.de/qdx/QDXComplaint_V2.0.xsd">
                ...
            </qdx:QDXComplaint>
        </qer:QDXEnvelopeResponse>
    </env:Body>
</env:Envelope>

--mime-boundary                                                    Attachment 1
Content-type: image/jpg
Content-ID: 1
Content-Transfer-Encoding: binary

...binäres jpg-Bild....

--mime-boundary                                                    Attachment 2
Content-type: video/mpeg
Content-ID: 2
Content-Transfer-Encoding: binary

...binäres mpeg-Video...
```

Fig. 4-10: SOAP message in multi-part MIME format

5 Appendix

5.1 WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:tns="urn:vda:qdx:QDXComplaintProcess_V2.0.wsdl"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:ns="urn:jai:qdxQDXEnvelopeRequest:2:0"
xmlns:ns1="urn:jai:qdxQDXEnvelopeResponse:2:0"
xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
name="QDXComplaintProcess"
targetNamespace="urn:vda:qdx:QDXComplaintProcess_V2.0.wsdl">
        <wsdl:types>
          <!-- Definition of request- and response-types-->
          <xsd:schema
targetNamespace="urn:vda:qdx:QDXComplaintProcess_V2.0.wsdl"
          elementFormDefault="qualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:envreq="urn:jai:qdxQDXEnvelopeRequest:2:0"
          xmlns:envresp="urn:jai:qdxQDXEnvelopeResponse:2:0">
                <xsd:import namespace="urn:jai:qdxQDXEnvelopeRequest:2:0"
                schemaLocation="../XSD/QDXEnvelopeRequest_V2.0.xsd"/>
                <xsd:import namespace="urn:jai:qdxQDXEnvelopeResponse:2:0"
                schemaLocation="../XSD/QDXEnvelopeResponse_V2.0.xsd"/>
                <xsd:element name="QDXComplaintListRequest"
type="envreq:QDXEnvelopeRequestType"/>
                <xsd:element name="QDXComplaintList"
type="envresp:QDXEnvelopeResponseType"/>
                <xsd:element name="QDXComplaintRequest"
type="envreq:QDXEnvelopeRequestType"/>
                <xsd:element name="QDXComplaint"
type="envresp:QDXEnvelopeResponseType"/>
                <xsd:element name="QDXAcknowledgeComplaint"
                type="envreq:QDXEnvelopeRequestType"/>
                <xsd:element name="QDXEnvelopeResponse"
type="envresp:QDXEnvelopeResponseType"/>
                <xsd:element name="QDXResetAcknowledgeStatusComplaint"
                type="envreq:QDXEnvelopeRequestType"/>
                <!-- Verwendung QDXEnvelopeResponse -->
                <xsd:element name="QDXReport8D"
type="envreq:QDXEnvelopeRequestType"/>
```

Fig. 5-1: WSDL Complaint Process Part 1

```
        <wsdl:message name="QDXReport8DRequest">
          <wsdl:part name="parameter" element="tns:QDXReport8D"/>
        </wsdl:message>
        <wsdl:message name="QDXReport8DResponse">
          <wsdl:part name="parameter" element="tns:QDXEnvelopeResponse"/>
        </wsdl:message>
        <wsdl:message name="QDXAcknowledgeReport8DRequest">
          <wsdl:part name="parameter"
element="tns:QDXAcknowledgeReport8DRequest"/>
        </wsdl:message>
        <wsdl:message name="QDXAcknowledgeReport8DResponse">
          <wsdl:part name="parameter"
element="tns:QDXAcknowledgeReport8D"/>
        </wsdl:message>
        <wsdl:portType name="QDXComplaintProcessPortType">
          <wsdl:operation name="getQDXComplaintList">
              <wsdl:documentation>Operation is used for requesting a list of
available QDXComplaints
              </wsdl:documentation>
              <wsdl:input message="tns:QDXComplaintListRequest"/>
              <wsdl:output message="tns:QDXComplaintListResponse"/>
          </wsdl:operation>
          <wsdl:operation name="getQDXComplaint">
              <wsdl:documentation>Operation is used for requesting a dedicated
QDXComplaint
              </wsdl:documentation>
              <wsdl:input message="tns:QDXComplaintRequest"/>
              <wsdl:output message="tns:QDXComplaintResponse"/>
          </wsdl:operation>
          <wsdl:operation name="postQDXAcknowledgeComplaint">
              <wsdl:documentation>Operation is used for acknowledging of a
QDXComplaint
              </wsdl:documentation>
              <wsdl:input message="tns:QDXAcknowledgeComplaintRequest"/>
              <wsdl:output
message="tns:QDXAcknowledgeComplaintResponse"/>
          </wsdl:operation>
          <wsdl:operation name="postQDXResetAcknowledgeStatusComplaint">
              <wsdl:documentation>Operation is used for resetting the processing
status of a QDXComplaint
              </wsdl:documentation>
              <wsdl:input
message="tns:QDXResetAcknowledgeStatusComplaintRequest"/>
```

Fig. 5-2: WSDL Complaint Process Part 2

```
        <wsdl:operation name="getQDXComplaint">
            <soap:operation soapAction="urn:vda:qdx:QDXComplaintRequest"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="postQDXAcknowledgeComplaint">
            <soap:operation
soapAction="urn:vda:qdx:QDXAcknowledgeComplaint"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="postQDXResetAcknowledgeStatusComplaint">
            <soap:operation
soapAction="urn:vda:qdx:QDXResetAcknowledgeStatusComplaint"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="postQDXReport8D">
            <soap:operation soapAction="urn:vda:qdx:QDXReport8D"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
```

Fig. 5-3: WSDL Complaint Process Part 3

5.2 Definition of new XSD documents

The following is a definition of the new QDX documents.

5.2.1 QDXAcknowledgeComplaint

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ackco="urn:jai:qdxQDXAcknowledgeComplaint:2:0"
targetNamespace="urn:jai:qdxQDXAcknowledgeComplaint:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXAcknowledgeComplaint schema for QDX-version 2.0
		This document ist used for acknowledgement of a QDXComplaint.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXAcknowledgeComplaint" type="ackco:QDXAcknowledgeComplaintType"/>
	<xsd:complexType name="QDXAcknowledgeComplaintType">
		<xsd:sequence>
			<xsd:element name="BuyerParty">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="ID" type="xsd:string"/>
						<xsd:element name="AdditionalID" minOccurs="0">
							<xsd:complexType>
								<xsd:simpleContent>
									<xsd:extension base="xsd:string">
										<xsd:attribute name="schemeAgencyID"
										type="xsd:string" use="required"/>
									</xsd:extension>
								</xsd:simpleContent>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Complaint">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="ComplaintItemID" type="xsd:string"/>
						<xsd:element name="RevisionID" type="xsd:string" minOccurs="0"/>
						<xsd:element name="RevisionDateTime" type="xsd:dateTime"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-4: XSD QDXAcknowledgeComplaint

5.2.2 QDXAcknowledgeReport8D

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ack8d="urn:jai:qdxQDXAcknowledgeReport8D:2:0"
targetNamespace="urn:jai:qdxQDXAcknowledgeReport8D:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXAcknowledgeReport8D schema for QDX-version 2.0
		This document ist used for acknowledgement of an 8D-Report.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXAcknowledgeReport8D" type="ack8d:QDXAcknowledgeReport8DType"/>
	<xsd:complexType name="QDXAcknowledgeReport8DType">
		<xsd:sequence>
			<xsd:element name="SellerParty">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="ID" type="xsd:string"/>
						<xsd:element name="AdditionalID" minOccurs="0">
							<xsd:complexType>
								<xsd:simpleContent>
									<xsd:extension base="xsd:string">
										<xsd:attribute name="schemeAgencyID"
										type="xsd:string"
										use="required"/>
									</xsd:extension>
								</xsd:simpleContent>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Report8D">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="RevisionID" type="xsd:string" minOccurs="0"/>
						<xsd:element name="RevisionDateTime" type="xsd:dateTime"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-5: XSD QDXAcknowledgeReport8D

5.2.3 QDXAcknowledgeReport8DRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ack8dreq="urn:jai:qdxQDXAcknowledgeReport8DRequest:2:0"
targetNamespace="urn:jai:qdxQDXAcknowledgeReport8DRequest:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXAcknowledgeReport8DRequest schema for QDX-version 2.0
		This document ist used for requesting the acknowledgement of an 8D-Report.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXAcknowledgeReport8DRequest"
	type="ack8dreq:QDXAcknowledgeReport8DRequestType"/>
	<xsd:complexType name="QDXAcknowledgeReport8DRequestType">
		<xsd:sequence>
			<xsd:element name="BuyerParty">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="ID" type="xsd:string"/>
						<xsd:element name="AdditionalID" minOccurs="0">
							<xsd:complexType>
								<xsd:simpleContent>
									<xsd:extension base="xsd:string">
										<xsd:attribute name="schemeAgencyID"
										type="xsd:string" use="required"/>
									</xsd:extension>
								</xsd:simpleContent>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Complaint">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="ComplaintItemID" type="xsd:string"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Report8D">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="RevisionID" type="xsd:string" minOccurs="0"/>
						<xsd:element name="RevisionDateTime" type="xsd:dateTime"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-6: XSD QDXAcknowledgeReport8Drequest

5.2.4 QDXEnvelope

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:env="urn:jai:qdxQDXEnvelope:2:0"
targetNamespace="urn:jai:qdxQDXEnvelope:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXEnvelope schema for QDX-version 2.0
		This envelope ist used for for any transmission (active communication).
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXEnvelope" type="env:QDXEnvelopeType"/>
	<xsd:complexType name="QDXEnvelopeType">
		<xsd:sequence>
			<xsd:any namespace="##other"/>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-7: XSD QDXEnvelope

5.2.5 QDXEnvelopeRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:envreq="urn:jai:qdxQDXEnvelopeRequest:2:0"
targetNamespace="urn:jai:qdxQDXEnvelopeRequest:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXEnvelopeRequest schema for QDX-version 2.0
		This envelope ist used for any request (passive communication).
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXEnvelopeRequest" type="envreq:QDXEnvelopeRequestType"/>
	<xsd:complexType name="QDXEnvelopeRequestType">
		<xsd:sequence>
			<xsd:any namespace="##other"/>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-8: XSD QDXEnvelopeRequest

5.2.6 QDXEnvelopeResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:envresp="urn:jai:qdxQDXEnvelopeResponse:2:0"
targetNamespace="urn:jai:qdxQDXEnvelopeResponse:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXEnvelopeResponse schema for QDX-version 2.0
		This envelope ist used for any response to a request (passive communication).
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXEnvelopeResponse" type="envresp:QDXEnvelopeResponseType"/>
	<xsd:complexType name="QDXEnvelopeResponseType">
		<xsd:sequence>
			<xsd:element name="Code">
				<xsd:simpleType>
					<xsd:restriction base="xsd:integer">
						<xsd:pattern value="[0-9]{3}"/>
					</xsd:restriction>
				</xsd:simpleType>
			</xsd:element>
			<xsd:element name="CodeDescription"/>
			<xsd:element name="CodeDetails"/>
			<xsd:any namespace="##other" minOccurs="0"/>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-9: XSD QDXEnvelopeResponse

5.2.7 QDXComplaintList

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:coli="urn:jai:qdxQDXComplaintList:2:0"
targetNamespace="urn:jai:qdxQDXComplaintList:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXComplaintList schema for QDX-version 2.0
		This document ist used for transmitting a list of retrievable QDXComplaints.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXComplaintList" type="coli:QDXComplaintListType"/>
	<xsd:complexType name="QDXComplaintListType">
		<xsd:sequence>
			<xsd:element name="ComplaintList" maxOccurs="unbounded">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="BuyerParty">
							<xsd:complexType>
								<xsd:sequence>
									<xsd:element name="ID" type="xsd:string"/>
									<xsd:element name="AdditionalID" minOccurs="0">
										<xsd:complexType>
											<xsd:simpleContent>
												<xsd:extension
												base="xsd:string">
													<xsd:attribute
													name="schemeAgencyID"
													type="xsd:string"
													use="required"/>
												</xsd:extension>
											</xsd:simpleContent>
										</xsd:complexType>
									</xsd:element>
								</xsd:sequence>
							</xsd:complexType>
						</xsd:element>
						<xsd:element name="Complaint">
							<xsd:complexType>
								<xsd:sequence>
									<xsd:element name="DocumentID"
									type="xsd:string"/>
									<xsd:element name="ComplaintItemID"
									type="xsd:string"/>
								</xsd:sequence>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-10: XSD QDXComplaintList

5.2.8 QDXComplaintListRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:colireq="urn:jai:qdxQDXComplaintListRequest:2:0"
targetNamespace="urn:jai:qdxQDXComplaintListRequest:2:0">
        <xsd:annotation>
                <xsd:documentation>
                QDXComplaintListRequest schema for QDX-version 2.0
                This document ist used for requesting the list of available QDXComplaints (QDXComplaintList).
                Schema definition created by Verband der Automobilindustrie e.V.
                </xsd:documentation>
        </xsd:annotation>
        <xsd:element name="QDXComplaintListRequest" type="colireq:QDXComplaintListRequestType"/>
        <xsd:complexType name="QDXComplaintListRequestType">
                <xsd:sequence>
                        <xsd:element name="BuyerParty" minOccurs="0">
                                <xsd:complexType>
                                        <xsd:sequence>
                                                <xsd:element name="ID" type="xsd:string"/>
                                                <xsd:element name="AdditionalID" minOccurs="0">
                                                        <xsd:complexType>
                                                                <xsd:simpleContent>
                                                                        <xsd:extension base="xsd:string">
                                                                                <xsd:attribute name="schemeAgencyID"
                                                                                type="xsd:string" use="required"/>
                                                                        </xsd:extension>
                                                                </xsd:simpleContent>
                                                        </xsd:complexType>
                                                </xsd:element>
                                        </xsd:sequence>
                                </xsd:complexType>
                        </xsd:element>
                </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

Fig. 5-11: XSD QDXComplaintListRequest

5.2.9 QDXComplaintRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:coreq="urn:jai:qdxQDXComplaintRequest:2:0"
targetNamespace="urn:jai:qdxQDXComplaintRequest:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXComplaintRequest schema for QDX-version 2.0
		This document ist used for requesting a dedicated and retrievable QDXComplaint.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXComplaintRequest" type="coreq:QDXComplaintRequestType"/>
	<xsd:complexType name="QDXComplaintRequestType">
		<xsd:sequence>
			<xsd:element name="BuyerParty">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="ID" type="xsd:string"/>
						<xsd:element name="AdditionalID" minOccurs="0">
							<xsd:complexType>
								<xsd:simpleContent>
									<xsd:extension base="xsd:string">
										<xsd:attribute name="schemeAgencyID"
										type="xsd:string" use="required"/>
									</xsd:extension>
								</xsd:simpleContent>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Complaint">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="ComplaintItemID" type="xsd:string"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-12: XSD QDXComplaintRequest

5.2.10 QDXResetAcknowledgeStatusComplaint

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:rasc="urn:jai:qdxQDXResetAcknowledgeStatusComplaint:2:0"
targetNamespace="urn:jai:qdxQDXResetAcknowledgeStatusComplaint:2:0">
	<xsd:annotation>
		<xsd:documentation>
		QDXResetAcknowledgeStatusComplaint schema for QDX-version 2.0
		This document ist used for resetting the acknowledge status of a QDXComplaint.
		Schema definition created by Verband der Automobilindustrie e.V.
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="QDXResetAcknowledgeStatusComplaint"
	type="rasc:QDXResetAcknowledgeStatusComplaintType"/>
	<xsd:complexType name="QDXResetAcknowledgeStatusComplaintType">
		<xsd:sequence>
			<xsd:element name="BuyerParty">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="ID" type="xsd:string"/>
						<xsd:element name="AdditionalID" minOccurs="0">
							<xsd:complexType>
								<xsd:simpleContent>
									<xsd:extension base="xsd:string">
										<xsd:attribute name="schemeAgencyID"
										type="xsd:string" use="required"/>
									</xsd:extension>
								</xsd:simpleContent>
							</xsd:complexType>
						</xsd:element>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
			<xsd:element name="Complaint">
				<xsd:complexType>
					<xsd:sequence>
						<xsd:element name="DocumentID" type="xsd:string"/>
						<xsd:element name="ComplaintItemID" type="xsd:string"/>
						<xsd:element name="RevisionID" type="xsd:string" minOccurs="0"/>
						<xsd:element name="RevisionDateTime" type="xsd:dateTime"/>
					</xsd:sequence>
				</xsd:complexType>
			</xsd:element>
		</xsd:sequence>
	</xsd:complexType>
</xsd:schema>
```

Fig. 5-13: XSD QDXResetAcknowledgeStatusComplaint

5.3 Status codes for QDXEnvelopeResponse

5.3.1 Status codes for a successful event

Status code	Status description	Significance
200	Request of QDXComplaintList succeeded	The call-up of the list of complaints to be collected was successful
201	Request of QDXComplaint succeeded	The call-up of a complaint was successful
202	Transmission of QDXAcknowledgeComplaint succeeded	The transmission of an acknowledgement of the processing of a complaint was successful
203	Transmission of QDXResetAcknowledgeStatusComplaint succeeded	The transmission of a reset of a processing status was successful
204	Transmission of QDXReport8D succeeded	The transmission of an 8D report was successful
205	Request of QDXAcknowledgeReport8D succeeded	The call-up of a processing confirmation for an 8D report was successful

Table 5-1: Status codes for a successful event

5.3.2 Status codes for an unsuccessful event

Status code	Status description	Significance
400	No QDXComplaints available	There are no complaints to be collected
401	The requested QDXComplaint is not available	The requested complaint is not available
402	Unknown customer identification	The customer number which has been given is not recognized
403	Unknown additional customer identification	The "additional customer number" which has been given is not recognized
404	Acknowledgement the specified QDXComplaint is not possible	It is not possible to acknowledge the complaint which has been entered
405	Unknown revision of the QDXComplaint	The version of the complaint is not recognized
406	Unknown revision date of the QDXComplaint	The version date of the complaint is not recognized
407	Unknown QDXReport8D	The 8D report which has been entered is not recognized
408	Unknown revision of the QDXReport8D	The version of the complaint is not recognized
409	Unknown revision date of the QDXReport8D	The version date of the complaint is not recognized

Table 5-2: Status codes for an unsuccessful event

5.4 Optional individual agreements between the parties

The following table sets out the individual agreements between parties applicable within the framework of this specification.

Subject	Restriction	Section
Specifying the role of the customer in the structure of the communication	active, passive	2
Specifying the communication procedure	OFTP, Webservice	3
Webservice : defining waiting times : - between setting up an 8D report and the call-up for a processing confirmation of the 8D report. (Unless otherwise stated, the processing confirmation must be available within 90 sec.). - between the resetting of the processing status of a QDXComplaint and the repeated call-up of the QDXComplaint (Unless otherwise stated, the complaint must be available within 90 sec.).	x seconds, minutes, hours	3.1
If no processing confirmation is available, the process must be defined between customer and supplier : - call up a second time - then escalation and bilateral clarification between supplier and customer (the supplier is responsible for this).		
Webservice : define the access intervals for call-ups and for setting QDX documents.	after x minutes, at specified times; time-window	3.1
Restrict attachments	Size; quantity	4.5
Coding procedure for attachments must be defined. (recommendation : Base64).		

Table 5-3: Individual agreements between the parties

5.5 Glossary

Term / expression	Significance
8D report	The 8D report is used to document the elimination of the defect by the supplier
Complaint	A complaint represents a deviation from the specification.
CAQ	**C**omputer **A**ided **Q**uality **A**ssurance
HTTP	**H**yper**t**ext **T**ransfer **P**rotocol
HTTPS	**H**yper**t**ext **T**ransfer **P**rotocol **S**ecure
Integration process	A cross-system process, where communication between the systems is based on XML messages
Communication system	See "Routing system".
MIME	**M**ultipurpose **I**nternet **M**ail **E**xtensions
Multipart MIME	A multi-part MIME message contains several parts (Bodyparts), in which data (including binary date) are transmitted.
OFTP	**O**DETTE **F**ile **T**ransfer **P**rotocol
Payload	Payload describes the effective data which are exchanged between two parties during a communication.
QDX	**Q**uality **D**ata e**X**change
Complaint process	A process in the automotive industry, where the customer complains to the supplier regarding deficiencies in parts (bought-in parts) which have been delivered and the supplier takes action to eliminate the deficiencies. These actions are documented as an 8D report
Routing system	A system which controls communication with internal and/or external systems. A routing system merely directs the data onward. No validation or data-processing is carried out.

SOAP	A network protocol for exchanging data on the basis of XML. SOAP originally represented **S**imple **O**bject **A**ccess **P**rotocol but this has not been used officially as an acronym since 2003
VDA	**V**erband **d**er **A**utomobilindustrie e.V. (German automotive industry association)
VDA-QMC	**Q**uality **M**anagement **C**entre of the **VDA**
WSDL	**W**eb**s**ervice **D**escription **L**anguage
XSD	**X**ML **S**cheme **D**efinition

Fig. 5-14: Glossary

5.6 Download

System files, examples of files and the WSDL can be downloaded from the following address :
http://vda-qdx.gefeg.com/

Quality management in the automotive industry

The current issues of VDA publications covering quality management in the automotive industry (QAI) can be viewed in the Internet under http://www.vda-qmc.de.

You can also place orders on this home-page.

Available from:

Verband der Automobilindustrie e.V. (VDA)
Qualitäts Management Center (QMC)
Behrenstraße 35
10117 Berlin
Germany
Telephone: +49 (0) 30 89 78 42-235
Telefax: +49 (0) 30 89 78 42-605
e-mail: info@vda-qmc.de,
Internet: www.vda-qmc.de

VDA QMC

Qualitäts Management Center
im Verband der Automobilindustrie