Common Integrative Implementation Guide to Supplement the EBICS Specification

(Electronic Banking Internet Communication Standard)

May 16th, 2011, based on EBICS-Version 2.5

Contents

1 Roadmap

1.1 Initial Situation

For the German as well as the French credit business, the EBICS specification (recent version is 2.4.1 including annexes 1 and 2) is regarded as the binding system of rules for the application of EBICS.

In principle, the specification applies to both countries likewise. In exceptional cases, however, a specific rule applying to one country may require an individual adaption of the specification. A functionality which is defined as optional may be supported (and mandatory) in one country and, at the same time, not be supported in another. In addition, an Implementation Guide for the introduction of EBICS in France has been developed reflecting special properties of the French market.

These deviations are due to different initial situations (predecessor systems) but do not constitute grave differences. Especially, the EBICS core functionalities are identical.

Therefore, it has to be emphasised that universally valid implementation standards will be possible after a transition period which is the unreserved intention of all participants. The participants, in this case, are the German banking sector represented by Zentraler Kreditausschuss (ZKA) and the French banking sector represented by Comité Français d'Organisation et de Normalisation Bancaires (CFONB).

A common Implementation Guide being likewise valid for Germany and France shall assist this process actively.

At present, the following documents exist as implementation aids:

1. The existing EBICS Implementation Guide (at present version 1.7) does not contain additional binding requirements and is only designated as an aid for the first steps in designing an EBICS product.
   In the past, it was used for the first implementations in Germany. Today, only the EBICS specification is applied.
   From the German as well as the French point of view, this IG is not regarded as binding.
2. The French community has developed a French IG (at present version 2.0) in which especially the French characteristics are described. In France, it is a necessary supplement to the specification.

1.2 Letter of Intent for a Common and Integrative Implementation Guide

In the conference of the EBICS COOP Expert Team on November 6th, 2009, it was unanimously recommended to constitute a common and integrative Implementation Guide as this is urgently required for further harmonising the EBICS standards. The objective target is to accomplish this common Implementation Guide together with EBICS version 2.5

The following procedure has been agreed on:

1. Development of a roadmap in which the deviations of the German and French layout of the EBICS standards are recorded. Each item is annotated with an approach for the consolidation which shall be scheduled as precisely as possible. Moreover, this document is a declaration of intent for the harmonisation of the specification layout and will be added as a management summary (introductory chapter) to the common Implementation Guide.

2. Development of common Implementation Guide as a binding requirement for the developers of EBICS products.
   Contents:

   a. Particular paragraphs of the EBICS Implementation Guide (Version 1.7)
      (a draft version describing which paragraphs are to be transfered is already on hand)
   b. Several paragraphs of the French Implementation Guide (Version 2.1)
   c. Additional topics

The outcome of 1. and 2. is the present document.

2 Common Implementation Guide

2.1 Introduction

This document, the “Common EBICS Implementation Guide” (common IG), is based on the “EBICS specification”. In addition to the stipulations and guidelines specified in this detailed concept, this guide provides information on selected aspects of the implementation process and will point out implementation alternatives. These recommendations are comprehensive and irrespective of the country where the implementation will be adopted.

The Common Integrative Implementation Guide has the following structure:

- Chapter 2.2 provides an introduction of the data model defined for EBICS.
- Chapter 2.3 shows implementation details for the identification and authentication signature. The structure of “XML signature” is handled in the same way as the allocation of the individual elements of this structure as expected in the EBICS context.
- Chapter 2.4 deals with the theme of key management, including the aspects of key generation and storage, the initialisation process and as well the use of certificates.
- Chapter 2.5 is concerned with information about the customer acknowledgement (customer protocol).
- Chapter 2.6 lists different technical clarifications/models.
- Chapter 2.7 is concerned with recommendations for client and server applications.
- Chapter 2.8 provides a collection of examples.
- In chapter 3 the current different usage of EBICS is illustrated.
- The common IG concludes with several annexes (illustrations and examples, chapter 4).

The target group for this document are developing vendors, banks and possibly corporate clients.

2.2 EBICS Entities

The basic terms to be defined for EBICS are:

- **Host (EBICS bank server):**
  The host is the EBICS bank computer system and it is identified by the HostID.
  It is also possible that banks have several HostIDs for their (several) EBICS servers. The financial institution communicates the EBICS HostID together with the URL for the bank access to the customer. In France the BIC is allocated to the HostID whereas in Germany the HostID usually is an 8 characters institution specific string (refer to chapter 3 “Different usage of EBICS” in this document).

- **Partner (or customer):**
  Organisational unit (company or individual) that concludes a contract with the bank. In this contract it will be agreed which order types (file formats) are used, which accounts are concerned, which of the customer's users (subscribers) communicate with the EBICS bank server and the authorisations that these users will possess. It is identified by the PartnerID.

- **User (or subscriber):**
  Human users or a technical system that is/are assigned to a customer. On the EBICS bank server it is identified by the combination of UserID and PartnerID.
  The technical subscriber serves only for the data exchange between customer and financial institution. The human user also can authorise orders.
  Remark: In the EBICS process/system "human" and "technical" differ from each other only in the point that for all EBICS requests, the technical subscriber allocates his subscriber identification to the field SystemID and generates the identification and authentication signature for the EBICS request.

- **Signature class:**
  At least one signature class is assigned to every user (subscriber) and relates to the ES (signature for authorisation). The signature class defines the quality of the subscriber's ES. It can be of type "E" (single signature), "A" (first signature), "B" (second signature) or "T" (transport signature).
  Detailed authorisation models can be defined individually for institutions, wherein the user gets different (i.e. more than one) signature classes with regard to the order type and/or the amount limit and/or the account used. In particular, this is possible if a bank only uses a subset of possible signature classes (e.g. French banks actually only use "E" and "T").
  Technical users (subscribers) are only assigned to signature class "T"
  (see also "User"). More details about signature classes see chapter 3.5.1 in the EBICS specification

- **Order Type / File Format:**
  The order type identifies the kind of EBICS transaction. There are two groups of order types:

  - *Organisational order types* (for the download of technical information, for the initialisation, for the key management, for the cancellation of orders and for the VEU)
    and the so-called
  - *bank-technical order types*: For the bank-technical order types a multitude of order types has been defined whereas the format is indicated by the order type identifier.
    An alternative is the use of two neutral order types (FUL for upload and FDL download). In this case the format identifiers are assigned by an additional EBICS parameter which is only defined for FUL and FDL. The different usage of order types in France and Germany see chapter 3 "Different usage of EBICS" of this document.

- **Order:**
  In EBICS every transmission from a customer to the bank (or vice versa) is called order

(upload order / download order). An unambiguous orderID is assigned to every upload order . This implicates that not only payment orders but also the upload of other files to the bank gets an orderID.
The application is to ensure the allocation of unambiguous orderIDs per each customer ID and per order type. The orderID especially serves the synchronizing of order data and electronic signatures but it also assigns a cancellation request to a certain order which is still not executed (in the case of VEU).

- **Contract:**
  The basic prerequisite for using EBICS is the conclusion of a **contract** between customer and bank. In this contract it will be agreed which order types the customer will conduct with the bank, which accounts are concerned, which of the customer's users work with the system and the **authorisations** that these users will possess. The contract reflects the agreed authorisations (authorisation/role model). The download of HKD/HTD gives information on the agreement between partner (customer) and bank. In the following diagram the date elements are shown (where each user of a partner has got n authorisations deduced from the contract between customer and bank):

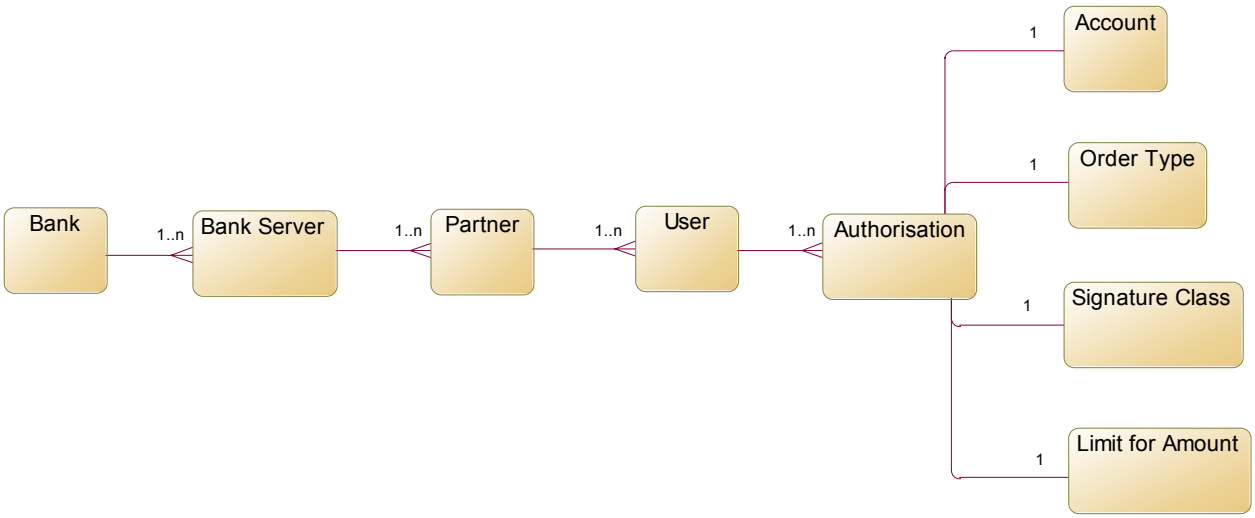

Diagram 1: Model of the relationship between user and authorisation(s)

The next diagram is an example showing a payment order with its basic properties (information). More details concerning the ES quantity see in EBICS specification chapter 11.2.3

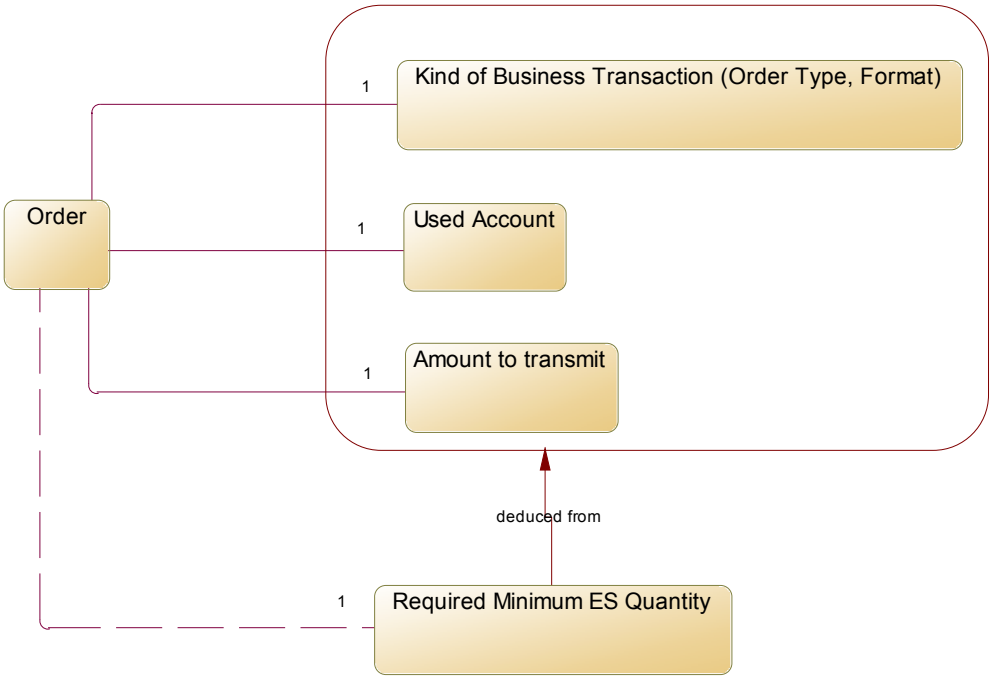

Diagram 2: Example for the relationship between payment order and minimum ES quantity

The necessary particular signature class for the above mentioned ES's depends on the authorisation model defined by each individual financial institution.

2.3 Implementation Instructions for the identification and authentication signature

The format "XML signature" in accordance with RFC 3275 is used for the EBICS identification and authentication signature in Version "X002". Detailed information are given in the EBICS specification document.

2.3.1 XML signature in the EBICS context

The EBICS structure element for the identification and authentication signature is called `ebics/AuthSignature` and can be found in the EBICS schemas "ebics_request_H004.xsd" und "ebics_response_H004.xsd" in an enumeration sequence between the EBICS header and body.

The EBICS schemas "ebics_keymgmt_request_H004.xsd" and "ebics_keymgmt_response_H004.xsd" are borrowed from the aforementioned standard EBICS schemas. They are used in the following key management order types:

- "INI" (subscriber initialisation: sending the public bank-technical key)

- "HIA" (subscriber initialisation: sending the public identification and authentication keys and the encryption keys)
- H3K (subscriber initialisation: sending the public bank-technical key, public identification and authentication keys and the encryption keys – only possible in the case of certificates)
- "HSA" (subscriber initialisation: sending the public identification and authentication keys and the encryption keys, bank-technically signed with the existing FTAM signature key)
- "HPB" (download the financial institution's public key)

For the EBICS key management schema "ebics_keymgmt_request_H004.xsd", only order type "HPB" requires an identification and authentication signature ("INI", "HIA", "H3K" and "HSA" do not), in the case of the EBICS key management schema "ebics_keymgmt_response_H004.xsd" the identification and authentication signature does not exist.

In the case of "INI", "HIA" and (for the direction of responses) "HPB", the necessary public keys of the customer and the financial institution for verifying an identification and authentication signature are not yet known or are not yet verified. Generation and verification of the identification and authentication signature of these order types is therefore pointless or even impossible.

2.3.2 Composition of the XML signature structure

The element `AuthSignature` is of type `ds:SignatureType` (with `xmlns:ds="`http://www.w3.org/2000/09/xmldsig#`"`), which has the following substructure:

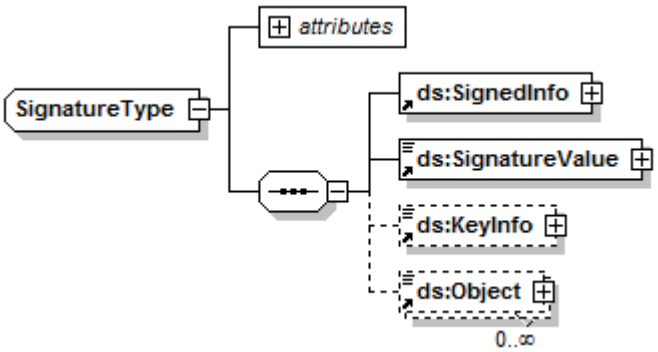

Diagram 3: Type "SignatureType" of the XML signature

The element `ebics/AuthSignature/ds:SignedInfo` contains parameters for the XML signature, but not the signature itself. This is stored in the element `ebics/AuthSignature/ds:SignatureValue`.

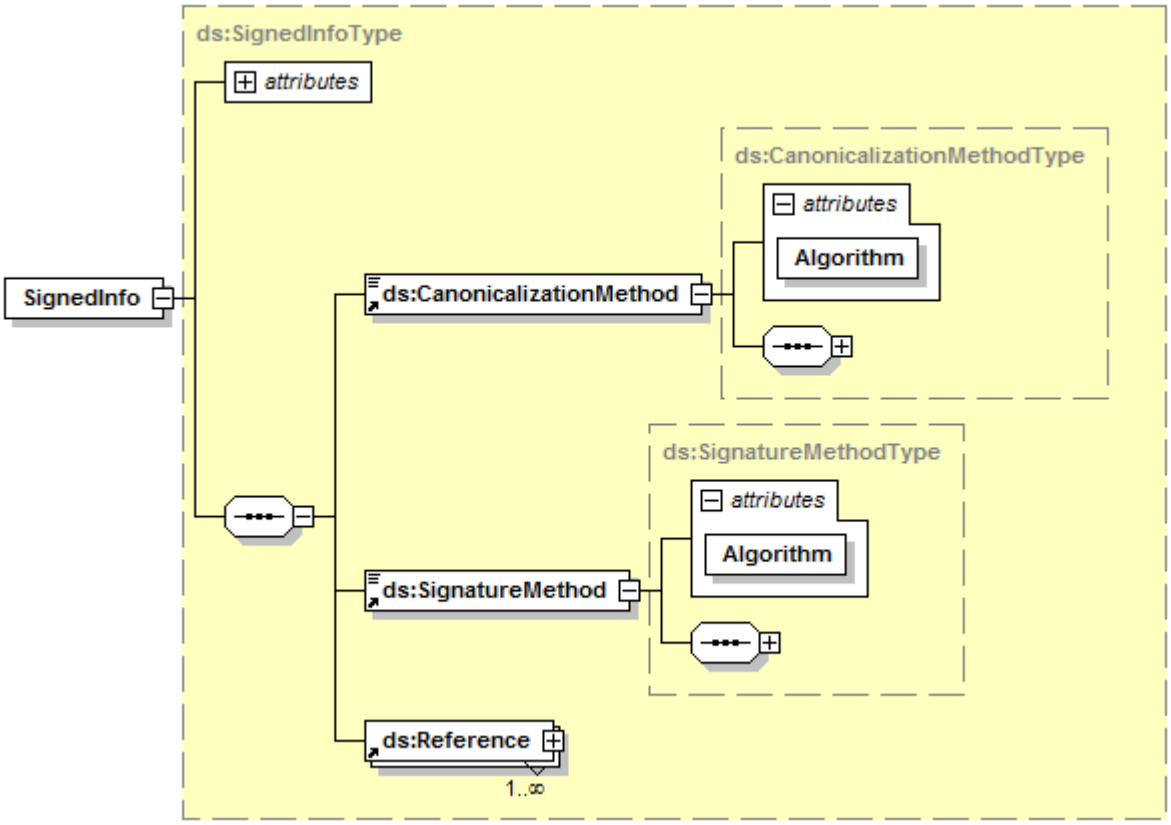

Diagram 4: Element "SignedInfo" of the XML signature (from "AuthSignature" of type "SignatureType")

The following parameters of `ds:SignatureType` are required in connection with EBICS:

- Canonisation algorithm (`ds:CanonicalizationMethod`): Here, the algorithm is specified in the attribute `@Algorithm`, in the form of a URI, that is to canonise the `ds:SignedInfo` structure itself, i.e. convert it into a standardised form, before it is used for signature configuration
- Signature algorithm (`ds:SignatureMethod`): Using the attribute `@Algorithm`, at this point the algorithm is specified in the form of a URI to configure and verify the signature.

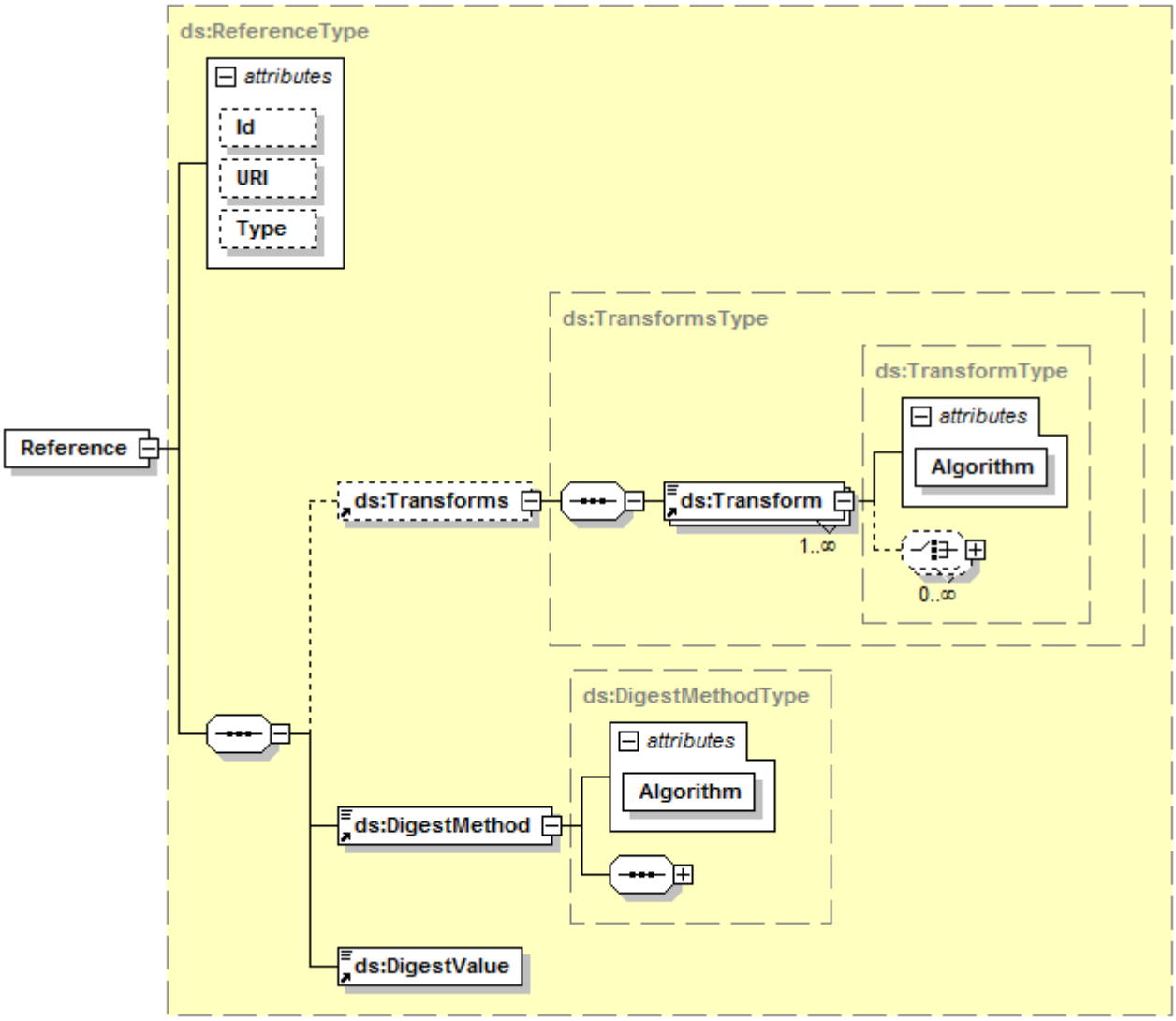

Diagram 5: Element “Reference” of the XML signature (from “SignedInfo”)

- Reference of the data that is to be signed (`ds:Reference`): Again, this element contains a substructure with the help of which a description can be given for the data that is to be signed and its hash value:

  `@URI` references the data structure that is to be signed.

  In the attribute `@Algorithm` of the multiple sub-element `ds:Transform`, `ds:Transforms` specifies, in the form of a URI, the algorithms with the help of which the referenced data are to be transformed before calculation of the hash value.

  In the attribute `@Algorithm`, `ds:DigestMethod` names, in the form of a URI, the algorithm for hash value configuration via the transformed data.

  `ds:DigestValue` finally contains the hash value for referenced, transformed data in base64 coding.

2.3.3 Allocation of the XML signature structure

The aforementioned XML fields for configuration of the identification and authentication signature in Version “X002” are to be allocated in accordance with the following specifications:

- `ds:CanonicalizationMethod@Algorithm`: The algorithm “Canonical XML“ (“http://www.w3.org/TR/2001/REC-xml-c14n-20010315“) is to be used here.
- `ds:SignatureMethod@Algorithm`: RSA with SHA-256 (“http://www.w3.org/2001/04/xmldsig-more#rsa-sha256“) serves as an algorithm pair for configuration of the identification and authentication signature.
- `ds:Reference@URI`: The reference, in the form of a URI, must cover all elements & their sub-structures that have occupied the attribute @authenticate with the value true. @URI="#xpointer(//*[@authenticate='true'])".
- `ds:Reference/ds:Transforms/ds:Transform@Algorithm`: “Canonical XML“ is again to be used as a transformation algorithm for the referenced XML data. Other transformations are not envisaged.
- `ds:Reference/ds:DigestMethod@Algorithm`: SHA-256 (”http://www.w3.org/2001/04/xmlenc#sha256“) serves as the hash algorithm for the transformed XML data.
- `ds:Reference/ds:DigestValue`: The result of the following operations is to be entered in this field:

  Transformation of the referenced XML data in accordance with “Canonical XML”

  Hash value configuration of the transformed XML data via SHA-256

  Encoding of the hash value with base64.

In Version “X002“ , the element `ebics/AuthSignature/ds:SignatureValue` contains the result of the following operations for the identification and authentication signature:

1. Canonisation of the `ebics/AuthSignature/ds:SignedInfo` structure in accordance with ”Canonical XML“
2. Hash value configuration via the canonised data with SHA-256
3. Signature computation via the calculated hash value with RSA: Here, the private identification and authentication key of the subscriber or – where available – the technical user is used in the case of EBICS requests, and in the case of EBICS responses the private identification and authentication key of the financial institution is used
4. base64-encoding of the signature.

The optional elements `ds:KeyInfo` and `ds:Object` in `ebics/AuthSignature` are not allocated:

- `ds:KeyInfo` can accept the public keys or certificates required for verification of an XML signature. With EBICS, these are already distributed and persistently stored at both the server's and client's end for key management order types "INI", "HIA", "HSA" and "HPB", and also additional order types for key alteration PUB and HCA
- `ds:Object` can contain a data object that is to be signed. However, within the framework of the EBICS identification and authentication signature, only those XML structured including sub-structures are signed that contain the attribute `@authenticate='true'` (namely the technical and order-related control data incl. preliminary checking data and transaction key information, in addition to the ES(s)).

Examples of EBICS messages with identification and authentication signature in accordance with the XML signature are given in the EBICS specification document (e.g. chapter 5.5)

For generation of the XML signature, see also:
http://www.w3.org/TR/xmldsig-core/#sec-CoreGeneration
For the canonisation algorithm "C14N", see also:
http://www.w3.org/TR/2001/REC-xml-c14n-20010315

2.3.4 Meaning of the XPointer expression in `ds:Reference@URI`

All elements of the EBICS message with the attribute `@authenticate='true'` and its sub-structures are referenced with the XPointer expression `"#xpointer(//*[@authenticate='true'])"` for the XML signature attribute `ds:Reference@URI` (see http://www.w3.org/TR/xmldsig-core/#sec-Same-Document).

XPointer is a standard (see http://www.w3.org/TR/WD-xptr) that expands XPath notation (see http://www.w3.org/TR/WD-xptr) with additional constructs for localisation within XML documents. Here, the term "`#xpointer`" introduces labelling of the URI as an expression in accordance with the XPointer standard. The entry "/ /*" is a shorthand form that initially selects all elements of the XML document. "`[@authenticate='true']`" is used to restrict this selection to those elements that have an attribute `@authenticate` with the value "true". XML signature defines that when using XPointer syntax, the sub-structures of the elements selected in this way are included in the signature.

In contrast to explicit listing of all (sub-) structures that are to be identified and authenticated, the above expression has the following advantages in the EBICS context:

1. Succinct short form: Although a number of structures (both in the header data as well as the body data) are to be identified and authenticated, a short line is sufficient to specify the scope of the data that has been signed or that is to be signed. Sub-structures are automatically included in accordance with XML signature.
2. Immediate documentation of the scope of the signature: The schema itself immediately defines the elements and sub-structures that have been signed or that

are to be signed via the attribute `@authenticate='true'`, whose appearance in the XML document is specified and documented directly in the XML schema definition. Therefore no separate concept document is required to check whether the specified data scope is covered by the signature; this task is already fulfilled in the schema validation.

3. Invariance in changes to the XML schema: Modifications to the XML structure (e.g. within the framework of a new EBICS version) do not influence the meaning of the `@authenticate` attribute values. Even if individual elements are re-named, removed, added or re-positioned, only the `@authenticate` attributes from the schema decide as to the scope of the data for signature configuration. On the other hand, if absolute paths were used in the URI reference specifications, in the event of any change the consistency of these would also have to be checked with regard to the changed XML structure.

2.3.5 Verification of the identification and authentication signature

The following steps are to be carried out for verification of the identification and authentication signature:

1. Validation of the EBICS message with regard to the matching EBICS schema. A positive result guarantees that the `@authenticate='true'` attributes are actually present at the places required in the schema.
2. Verification of the composition of the XML signature structure . Here, care should be taken that the reference URI and the canonisation and transformation algorithm corresponds with the specifications in Chapter 2.3.3 and that no additional transformation algorithms are entered. Deviating or additional transformation algorithms can have an influence on the type and scope of the data that is to be signed and can hence distort the results of the identification and authentication check.
3. Configuration of the hash value via the elements recorded by the reference URI in accordance with the specifications returned in `ebics/AuthSignature/ds:SignedInfo/ds:Reference` and comparison with the base64-decoded value of the returned element `ebics/AuthSignature/ds:SignedInfo/» ds:Reference/ds:DigestValue`. In the event of a difference between these values, the identification and authentication check is deemed to have failed.
4. Calculation of the signature hash value in accordance with the specifications in `ebics/AuthSignature/» ds:SignedInfo`, i.e. execution of operations 1 and 2 to allocate the element `ebics/AuthSignature/ds:SignatureValue` from Chapter 2.3.3. After successful checking of steps 1 to 1 of this verification procedure, the data from

the XML signature structure is identical to the specifications of Chapter 2.3.3 for generating the identification and authentication signature.

5. Execution of RSA signature verification using the public RSA identification and authentication key of the other party on the base64-decoded version of the supplied signature value in the element `ebics/AuthSigature/ds:SignatureValue`. In the case of EBICS requests, the financial institution determines the identity of the other party via the control data (subscriber ID or technical system ID and customer ID), in the case of EBICS responses the customer system selects the appropriate key of the contacted financial institution. The result of the RSA operation is the signature hash value calculated by the other party.

6. Comparison of the two signature hash values. The identification and authentication signature is only successfully verified if the hash values are identical, i.e. the elements and sub-structures of the EBICS message with the attribute `@authenticate='true'` are authentic and the identity of the sender is assured.

For more details on the validation process of XML signatures, see:
http://www.w3.org/TR/xmldsig-core/#sec-CoreValidation

2.4 Key Management and Use of Certificates

2.4.1 Key storage

Storage of the keys must guarantee the integrity of the public keys and the integrity and secrecy of the private keys, both in the customer system and the bank system. Components that are responsible for storage of the keys are generally referred to as keystores. Keystores can be implemented as

- components of special crypto-hardware (Smartcards, HSM (Hardware Security Modules), USB tokens). In order to attain the independence of special hardware, it is important to implement access to the hardware via standard interfaces
- purely software components (software keystores). If different applications have to share a keystore, it is expedient to store the objects of the keystore such as keys, certificates, etc. directly in a standard format or at least to be able to export these in a standard exchange format.

Standard formats for the storage of keys are:

2.4.1.1 PKCS#12

The standard PKCS#12 [1] (Personal Information Exchange Syntax) v1.0 defines a secure exchange and storage format for both private keys and X.509 certificates. In this connection, secure means that both the integrity and the secrecy of the private key or the certificate is supported.

PKCS#12 ensures security of integrity and secrecy on the basis of passwords. A symmetrical key is derived from a (secret) password that is used for the encryption of data (e.g. the private key) whose secrecy is to be ensured. Analogously, a key is generated from a password that then flows into the calculation of the MACs (Message Authentication Code) of the data (such as e.g. certificates) whose integrity is to be ensured.

PKCS#12 is based on the standard PKCS#5 [2](Password-Based Encryption Standard) v2.0 which defines the process for the derivation of symmetrical keys from passwords. Furthermore, PKCS#12 is based on the standard PKCS#8 (the private-key information syntax standard) v1.2 which defines the format of secret keys.

PKCS#12 does not explicitly support the representation of public keys. Nevertheless, the standard allows password-based secure representation of any data types.

2.4.1.2 XML

The following aspects of the XML standard can be used for the storage of public keys:

- The standard “XML signature“ defines the (complex) XML type `KeyInfo` that allows the representation of public RSA keys as a combination of modulus and exponent (XML type `RSAKeyValue`) or as an X509 certificate (XML type `X509Data`).
- XML signature supports hash MACs as a signature algorithm, especially also HMAC SHA-1.
  These MACs can thus be used to ensure the integrity of XML elements on the basis of passwords.

On the other hand, there is no separate pre-defined XML type for representation of the private RSA keys.

EBICS provides for the transportation of public keys between the customer system and the bank system.

[1] References: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf, ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12-tc1.pdf

[2] References: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf, ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.asn

The order data for HIA, H3K, HSA, HPB or HCA orders are defined with the help of XML types `HIARequestOrderData`, `H3KRequestOrderData`, `HSARequestOrderData`, `HBPResponseOrderData` and `HCARequestOrderData`. In each case, these XML types contain elements of type `RSAKeyValue` or `X509Data`.

For the order data of INI or PUB orders the element structure `SignaturePubKeyOrderData` has to be used.

2.4.2 Use of Certificates (in France)

EBICS transaction can base on the use of X509 certificates (this is the case in France).
EBICS customers have the choice of two kinds of certificates:

1. self-signed certificates (generated by the user/customer himself)
2. certificates issued by a CA

The bank server needs three public subscriber keys, each for authentication, encryption and the ES. Each public key is stored in a certificate on a hardware or software storage medium. Each of the three certificates for the three use cases is transmitted to the server in an initialization file using the EBICS protocol.
As to the validation process, the case of using a self-signed certificate differs from the case of a certificate delivered/issued by a CA. If **using a self-signed certificate**, the validation is not possible by the certification chain. The authentication must be ensured by a second mechanism which is different from the initialization file generated by the customer system. In this case, the authentication is achieved by sending a confirmation to the bank (while at the same time sending the certificate via EBICS) but by way of another channel ("INI-letter" as described in the EBICS specification). How to send this confirmation must be stated in the contract between customer and bank.
Optionally, if proposed by the bank, this certificate confirmation file can be transmitted by another secured electronic channel (different from EBICS).

If using a **certificate issued by a CA**, the control of the certificate's certification chain allows a complete check of the certificate. No "INI-letter" is needed in that case (for more details refer to chapter 2.4.3.2).

In France, the RSA key length must be at least 2048 bits. However, the client has to check the interoperability with his bank if he uses a key length of more than 2048 bits.
In France, the signature algorithm for the CA signature is RSA-SHA2 (or SHA1, but only until 2013).

If certificates delivered by a CA are revoked the certificate cannot be used for EBICS transactions any longer.
Each certificate has got a validity period (date of expiration). Before its expiration it is necessary to obtain another valid certificate.
Therefore, the integral part in the signature verification process is the status check of the certificate, namely the validity and a possible revocation of the certificate. A revocation can be checked for example by the "Certificate Revocation List".

If the check fails the certificate must not be used any more and the user is assigned the status “revoked by user” (error code EBICS_INVALID_USER_STATE).

2.4.3 Initialisation

In EBICS, the user (subscriber) can be initialised as follows:

2.4.3.1 INI / HIA (initialisation with letters)

The two separate orders, INI and HIA transmit the user’s public keys (INI for the ES, HIA for the keys for identification and authentication as well as encryption). The authentication must be ensured by a second mechanism which is different from the initialization file generated by the customer system.
INI / HIA has to be chosen if

1. no certificates are used
2. or – in the case of certificates - the key for the ES does not base on a certificate issued by a CA
3. in case of certificates issued by a CA, H3K is recommended, but INI/HAI is an alternative if the H3K process fails

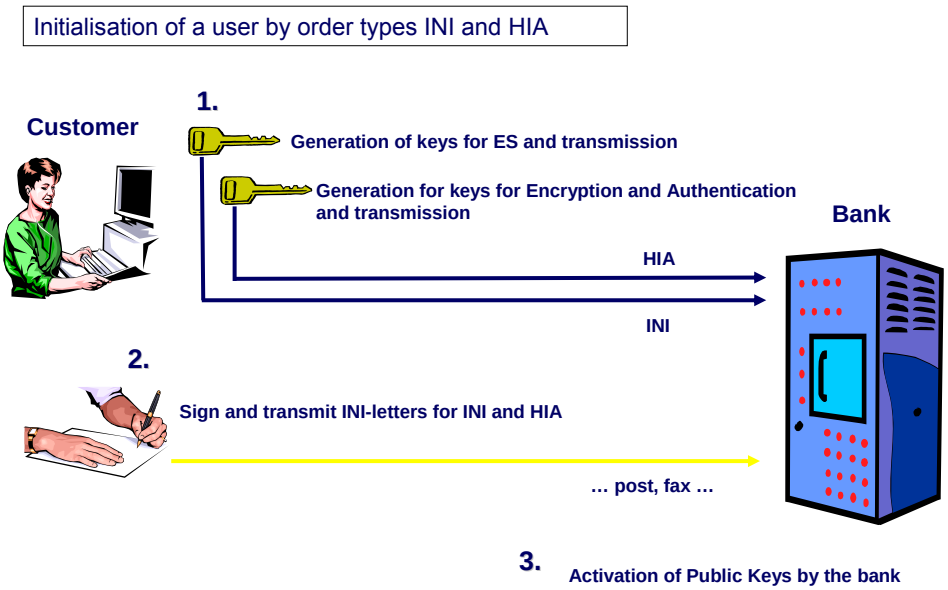

Diagram 6: Initialisation by INI / HIA

For more details about the initialization refer to the EBICS specification.

2.4.3.2 H3K (initialisation without letters)

When using certificates the initialisation can be executed in one step. However, the basic requirement is that for the ES key a certificate issued by a CA is available. In this case no letter for INI is necessary.

1. The upload of the public keys for ES (authorisation), encryption and authentication are sent to the bank by the order type H3K. The H3K-Request is already signed by an ES (using the (private) ES key).
2. As the ES key bases on a certificate issued by a CA the letter for HIA is also not necessary. Nevertheless, checks on the bank's side (before using the keys the first time) are necessary:
    a. Does an agreement for the use of the certificate exist?
    b. Are the administration steps for the customer/user finalized at the EBICS server and is the user known at the EBICS server?
    c. Is the certificate valid?

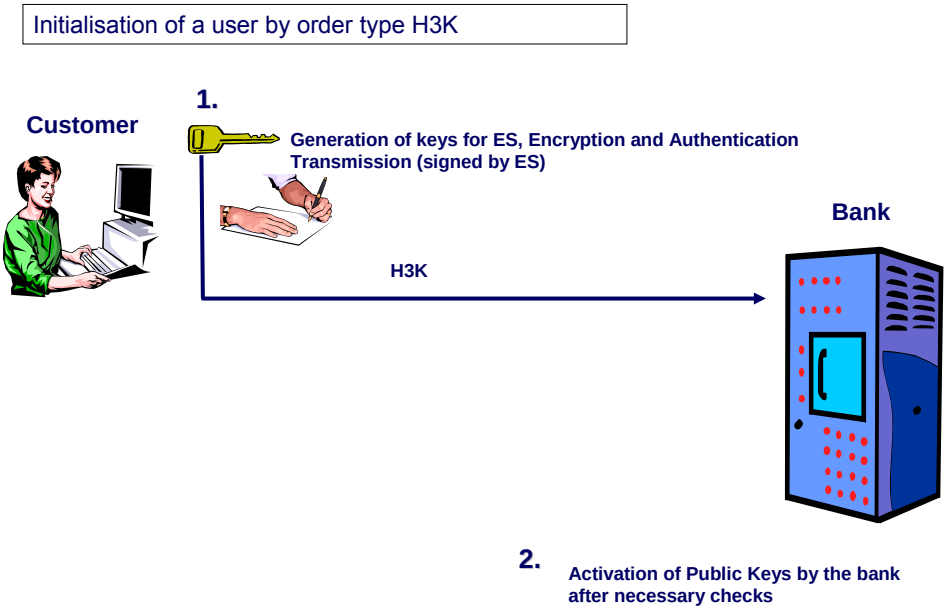

Diagram 7: Initialisation by H3K

For more details about the initialisation refer to the EBICS specification.

2.4.4 Verification of the bank keys

A prerequisite for the transmission of orders via EBICS is the download of the bank key via EBICS using HPB, followed by a verification of this bank key. For this reason, the financial institutions provide the bank keys via EBICS while their hash values are provided via a second communication channel that is independent of EBICS.
The customer system must prompt the subscriber to verify the keys that are downloaded via HPB. The customer system must calculate the hash values of this key in order to allow a comparison with the hash values provided by the bank (e.g. published in the portal).

If the hash value comparison is carried out manually by the subscriber, the associated subscriber must confirm the positive comparison.

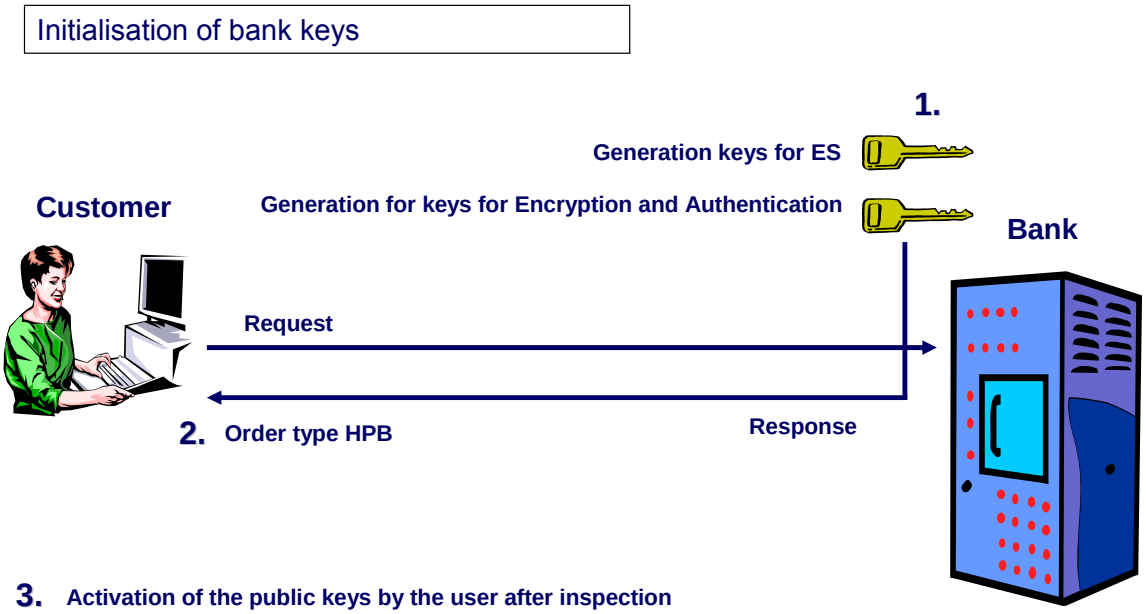

Diagram 8: Initialisation of bank keys

2.4.5 Amendment of the subscriber keys

For the amendment of keys the order type HCS (amendment of all keys in one step) is recommended (defined as from schema version H003).

An alterative is the use of HCA and PUB:
The EBICS customer software should present the amendment of the subscriber key as a technical procedure. The division into individual orders HCA and PUB cannot be concealed from the subscriber since both orders require the subscriber's deliberate signature. The following tables clarify which public/private subscriber keys are used, depending on the sequence, for processing HCA and PUB. Here, transmission without technical subscribers is considered.

Sequence of processing of order type			**old** subscriber key		**new** subscriber key	
			private	public	private	public
1.	HCA	Order data				☒
		ES	☒			
		Identification and authentication signature	☒			
2.	PUB	Order data				☒
		ES	☒			
		Identification and authentication signature			☒	

Sequence of processing of order type			**old** subscriber key		**new** subscriber key	
			private	public	private	public
1.	PUB	Order data				☒
		ES	☒			
		Identification and authentication signature	☒			
2.	HCA	Order data				☒
		ES			☒	
		Identification and authentication signature	☒			

If the transmission of the PUB and the HCA order takes place via a technical subscriber, in each case the identification and authentication signature is formed with the identification and authentication signature of a technical subscriber but it may be advantageous to carry out HCA before PUB. This is particularly the case when e.g. the updating of the subscriber key is associated with the renewing of the subscriber's chip card. Both orders are then signed with the subscriber's old bank-technical key and are thus signed with the same chip card.
In general, the sequence of HCA and PUB should be selected so that the number of changes between the old and the new key that are visible to the subscriber are kept to a minimum.

2.4.6 Compendium: upload, download and distributed electronic signature (VEU)

In the following diagrams the course of the basic actions concerning upload, download and the distributed electronic signature is shown:

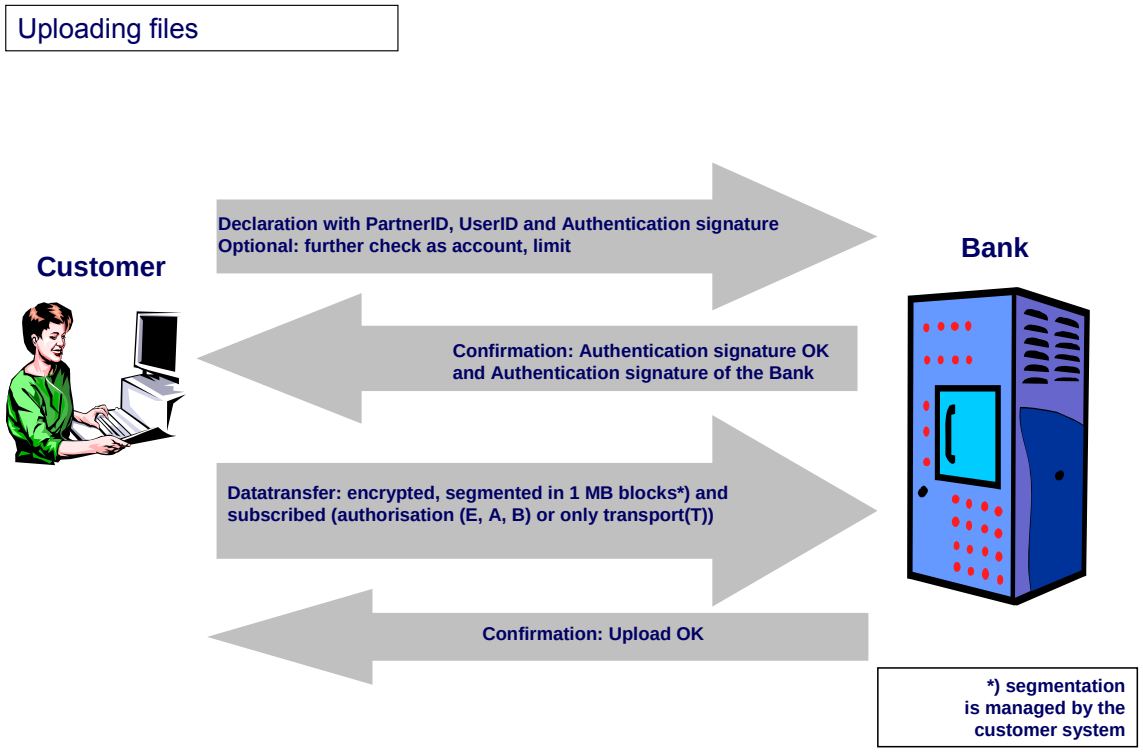

Diagram 9: Upload process

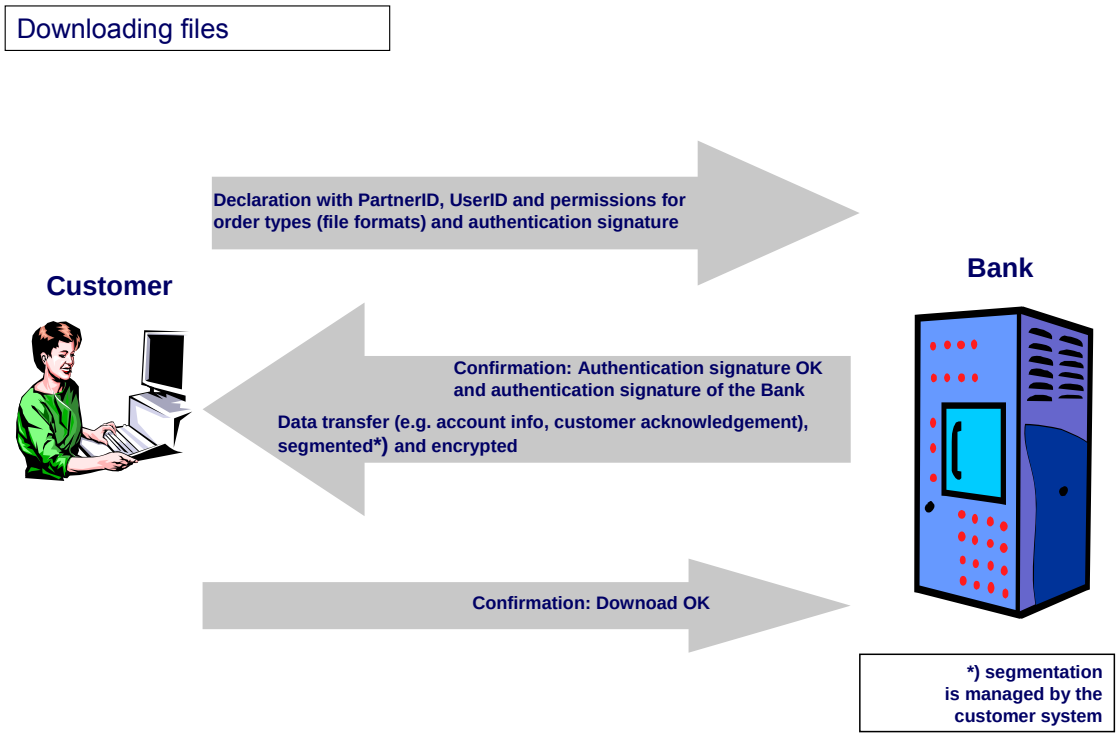

Diagram 10: Download process

VEU – basic actions:

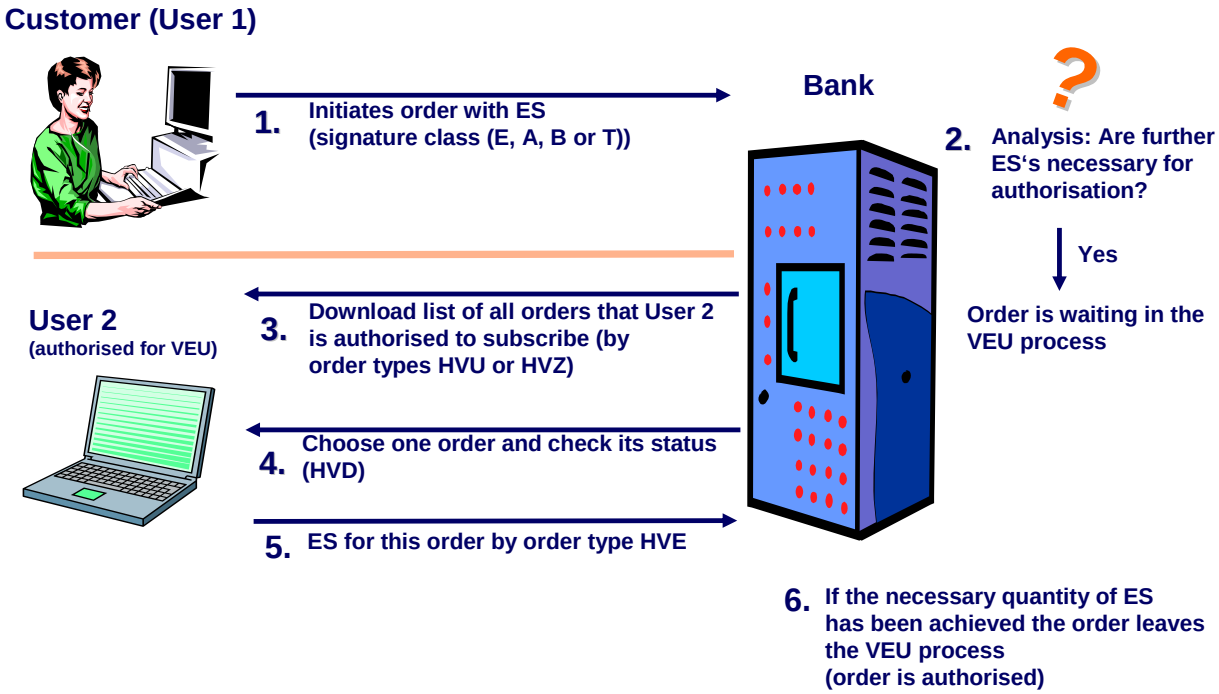

Diagram 11: Distributed electronic signature (VEU)

2.5 Acknowlegement for the customer

The acknowlegement (protocol) for the customer gives information about all actions and results that occur while uploading, downloading, or signing files and may give – in addition – information about the content of the order/file (display file).

German banks still support the order type PTK whereas French banks support FDL with a certain format (ACK). More information on these previous formats can be taken from the annex (chapter 4) of this document.
With EBICS version 2.5 a consolidated format for the customer acknowledgement including a common subset of allocation rules is introduced which is described in chapter 10 of the EBICS specification.

2.6 Technical Clarifications

2.6.1 Replay avoidance using Nonce and Timestamp

2.6.1.1 Formats of “Nonce” and “Timestamp”

2.6.1.1.1 Format of “Nonce”

“Nonce“ is to be fully allocated, including leading nulls if necessary, with a cryptographically-strong random number with a size of 128 bits in hexadecimal notation (32 digits from 0-9 and capital letters A-F in accordance with the canonical form of the XML schema type “hexBinary”). The selection of a random number of this size ensures that the probability of a conflict occurring between the “Nonce” values of transactions that are being executed in parallel (even those of other subscribers/customers) is sufficiently small (see Chapter 2.6.1.2.2).

2.6.1.1.2 Format of “Timestamp”

A time stamp in “ISO 8601“ notation in accordance with the XML schema data type “dateTime” is to be used in a combined form comprising date and time to allocate the “Timestamp”. For the time zone, either UTC (Coordinated Universal Time, earlier called GMT) can be used (correspondingly, “Z” for “Zulu” is to be appended as a time zone marker) or the difference in comparison with UTC can be appended. In the latter case, aspects such as summertime must be taken into consideration.

In the EBICS context, the format of time stamps in accordance with ISO 8601 for combined specification of date / time is as follows:
CCYY-*MM*-*DD*T*hh*:*mm*:*ss*.*ttt*Z for UTC, or *CCYY*-*MM*-*DD*T*hh*:*mm*:*ss*.*ttt*±*OO*:*oo*
in the case of time specifications that deviate from UTC. Here, characters marked in bold are to be taken over unchanged (“±” means “either **+** or **–**“), the italic letters are wildcard characters:

CC for the century,

YY for the year of the century,

MM for the month of the year (01 for January),

DD for the day of the month,

hh for the hour of the day (24-hour clock format, e.g. 15 for 3pm),

mm for the minutes of the hour,

ss for the seconds of the minute,

ttt for the thousandths of the second,

OO for the difference in hours to UTC,

oo for the difference of the minute proportion to UTC,

2.6.1.2 Actions of the customer system

2.6.1.2.1 Allocation of “Nonce” and “Timestamp”

In the case of initial key management EBICS requests (i.e. only for order types “INI“, “HIA“, “HSA“ and “HPB“), the fields “Nonce“ and “Timestamp“ are only to be allocated if an identification and authentication signature is required in the request for the selected order type.

An example of syntactically-correct setting of the values “Nonce” and “Timestamp” is shown in the following XML excerpt:

```
<?xml version="1.0" encoding="UTF-8"?>
<ebics
 […]
 <header authenticate="true">
   <Nonce>01A56FF768B3B36C5120E9904A7FB035</Nonce>
   <Timestamp>2010-05-20T17:07:34.123+02:00</Timestamp>
   […]
 </header>
 […]
</ebics>
```

Further information on correct setting of the two XML schema elements can be found under http://www.w3.org/TR/xmlschema-2/#hexBinary (hexBinary) and http://www.w3.org/TR/xmlschema-2/#dateTime (dateTime).

2.6.1.2.2 Behaviour in the event of error message „EBICS_TX_MESSAGE_REPLAY“

The bank system uses the technical error code EBICS_TX_MESSAGE_REPLAY to signal that the EBICS message that has just been sent by the customer system contains a “Nonce” value that corresponds with one stored in the bank system, or that the “Timestamp” lies outside the tolerance period.

The use of cryptographically-strong random numbers as ”Nonce” practically excludes the coincidental occurrence of such a situation. With the specified Nonce length of 128 bits, the probability of any conflict between two independently-generated random Nonces is precisely 2^{-64}, that is, on average a conflict of this nature will occur once in approx. $1.845 \cdot 10^{19}$ cases. In addition, this conflict would have to occur within the tolerance period, usually a few hours.

Therefore after receipt of the report EBICS_TX_MESSAGE_REPLAY, the customer system must take into account the possibility of a replay attack, an intolerably-imprecise clock setting in the customer system or the bank system, or an error in its own transaction management in the assignment of “Nonce” values.

If the subscriber would nevertheless like to successfully transmit the EBICS message in question, they must at least first regenerate the fields ”Nonce“ and ”Timestamp“ in accordance with Chapter 2.6.1.1. In addition, the identification and authentication signature

must be regenerated since it also includes the two fields ”Nonce“ and ”Timestamp“ in the signature.

2.6.1.3 Actions of the bank system

2.6.1.3.1 Checking “Nonce” and “Timestamp”

When the bank system receives an initial EBICS message from a subscriber, it must carry out the following actions to check for message replay.

1. Validation of the formats of the received technical EBICS header data ”Nonce“ and ”Timestamp“ within the framework of schema validation. For standard EBICS requests, validation is carried out against the schema “ebics_request_H004.xsd”. Validation is carried out against the schema “ebics_keymgmt_request _H004.xsd” in the case of key management EBICS requests.
   If the schema check proves to be negative, the message has not been constructed in accordance with EBICS guidelines. Therefore the bank system must reply with the technical error code “EBICS_INVALID_REQUEST“.
   If a request has been received within the framework of key management for which a “Nonce” / “Timestamp” entry is not required, the replay check is dispensed with.
2. Comparison between the received “Timestamp” and the local time stamp:
   Normalised to UTC, the received “Timestamp” must be within the tolerance period that is stretched around the current time stamp of the bank system. This tolerance period will compensate for differences in precision between the clocks involved in the systems and possibly also early/late changeover to summer/wintertime. At the same time, the tolerance period determines when the bank system can delete stored “Nonce”/”Timestamp” pairs. An incoming message with a “Timestamp“ outside the tolerance period would not be accepted, therefore the stored “Nonce”/”Timestamp” pairs with a “Timestamp” outside the tolerance period are superfluous and can be deleted.
   The tolerance period must be set as a one-off occurrence by the bank system. The setting “± 6 hours” can serve as a concrete guideline value. Higher values (= large tolerance periods) increase the storage requirements for valid “Nonce”/“Timestamp” pairs, lower values (= smaller tolerance periods) increase the risk of rejected EBICS messages as a result of excessive clock differences between customer & bank systems.
   If the received “Timestamp” is not within the tolerance period there is a risk of message replay. Therefore the bank system must reply with the technical error code “EBICS_TX_MESSAGE_REPLAY”.
3. Comparison of the received “Nonce” with the “Nonce” values stored in the bank system. All “Nonce”/“Timestamp” pairs that originate from valid EBICS requests within the tolerance period are stored at the bank’s end. If the received “Nonce” corresponds with a stored “Nonce” the bank system must reply with the technical error code “EBICS_TX_MESSAGE_REPLAY”.

If the above checks are all passed, there is no message replay. The bank system can then proceed with the actions in Chapter 2.6.1.3.2.

2.6.1.3.2 Storage of “Nonce”/”Timestamp” pairs

If the received “Nonce”/“Timestamp” pair is successfully checked, the bank system must store this pair for later checks on other EBICS messages. If a database is used for this purpose, the “Nonce” field can be used as an indexed primary key of a database table, which has a number of advantages:

- “Nonce is a non-empty alphanumeric value of fixed length (representation as ”hexBinary“ corresponds to a string of length 32, formed from the characters “0”-“9” and “A”–“F”) that unambiguously labels the message. Hence it is suitable for use as a primary key.
- Later access in the framework of conflict checks always takes place on the basis of the similarity check. Therefore an index to this field means a substantial speed increase for such tests.

Other primary keys are not provided, i.e. storage of the “Nonce”/”Timestamp” pairs takes place throughout the bank system.

The “Timestamp“ field is required to restrict the validity of the “Nonce” field. Range comparisons are used for this task (e.g. “Timestamp” ≤ “current timestamp” + 6 hours and “Timestamp” ≥ “current timestamp” - 6 hours).

The bank system’s protection from manipulation of the data in connection with message replay is decisive for the validity of the above EBICS check. To achieve this, the following data and system components must imperatively be secured against compromise at the bank system’s end:

- The stored “Nonce”/“Timestamp” pairs
- The “Nonce”/“Timestamp” pairs of the EBICS request message that is to be checked
- The bank system’s internal clock
- The deletion process instructions.

2.6.1.3.3 Further Recommendations

DoS attacks by registered users/clients: At present, there is no limitation on a customer's number of sessions. Thus, a single customer may (accidentally) open numerous sessions. With version 2.1 of the EBICS detailed concept a corresponding error code has been introduced for banking systems wishing to impose a limitation on https sessions (09-1-1-19 (EBICS_MAX-TRANSACTIONS_EXCEEDED). A maximum number of parallel https sessions for each customer ID can be specified. The maximum number of sessions per customer can be parameterized.

2.6.2 Random Numbers

Cryptographic (or cryptographically-secure) **pseudo-random number generators** (PRNG) are deterministic algorithms via which a sequence of **non-predictable** random numbers can be generated based on a real random number as a starting value (seed). In this context, non-predictable means that no further random numbers that have already been generated by the PNRG in the past or that will be generated in the future can be derived from knowledge of a few already-generated random numbers.

Cryptographically-strong (or cryptographically-secure) **pseudo-random numbers** are generated by cryptographic PRNGs.

With EBICS, cryptographically-strong random numbers are used in transaction management and in the avoidance of replay: both transaction IDs and Nonces are cryptographically-strong random numbers with a length of 128 bits. The entropy of the utilised seed should be at least 100 bits. See also the regular publications of the “Overview on suitable algorithms” from the Regulatory Authorities for Telecommunications and Posts.

General recommendations for the generation of the random seed, also using computer hardware, can be found in RFC 1750 (Randomness Recommendations for Security). In particular, examples of non-secure PRNGs are also given here.

2.6.3 Character set

The valid allocation of the data elements in EBICS requests and EBICS responses are defined by the EBICS schema (xsd). The correct allocation can be checked by an XML parser.
Remark:

- The specific German characters ä, ö, ü and ß (upper and lower case) are excluded.
- The specific French characters and characters with accentuation signs (e.g. ç, œ é, è, à) are excluded (in upper and lower case).

The valid characters used in the exchanged files are defined in the respective message specifications.

2.7 Recommendations for clients and bank servers

2.7.1 Minimum requirements

Both customer system and bank system must comply with the requirements in the EBICS specification.
An important precondition is an access to a network based on IP.
It should be designend for the requirements concerning security, expected quantity and performance.

Before the implementation of the connection between a customer system and a bank server a contract must be signed between the customer and the bank for the government of the communication between bank and customer.
In this contract requirements concerning security, data storage etc. can be governed.

2.7.2 Further requirements

2.7.2.1 Actions of the customer system

Suspension request:
After submitting a suspension the subscriber will not able to effect any activity concerning remote data transmission any more. If the subscriber wants to conduct any finishing operations, e.g. cancellations, these have to be accomplished beforehand. The suspension can only be revoked by a reinitialisation. At the request of the customer, the bank may also revoke the suspension.

2.7.2.2 Actions of the bank system

Order of procedures within VEU
The bank system has to verify the correct order of the procedures within the VEU. Especially, HVU (or HVZ respectively) is a necessary precondition for the following steps.

Handling of orders not completely authorized
Orders which are not completely authorized are to be deleted after the period of time that has been agreed upon with the customers.

2.8 Examples

2.8.1 Workflow for A005

In www.ebics.org an example is provided for test purposes. It illustrates the cryptograhpic aspects – that is the canonicalization of the XML, calculation of hash codes and signatures, encryption and decryption.

2.8.2 Test Mode (use in France)

In France the EBICS standard has been newly introduced. So a test of the file transfer is required before it is used in production. Therefore, each server must be able to receive test and production files.
It has to be agreed in the contracts between customer and bank that the customer will get a test mode in his EBICS system. The test activities are as follows:

- The customer system must have a set-up which allow switching from test mode to production mode. Its use can only be made at the initiative of the customer in agreement with its bank.
- In order to avoid potential confusion, a manual intervention from the bank server side is unwanted.
- Test and production files differ in the parameter called "TEST" .
  If the value is “True” it is a test file. The absence of the set-up means a production file. The « TEST » indication must be included in the tag «OrderParam » in the following way:

```
<FULOrderParams>
  <Parameter>
    <Name>TEST</Name>
    <Value>TRUE</Value>
  </Parameter>
  <FileFormat
CountryCode="FR">pain.xxx.cfonb160.dct</FileFormat>
</FULOrderParams>
```

In www.ebics.org an example is provided for test purposes.

2.8.3 Examples for the customer acknowledgement

XML-examples for the use of the EBICS customer acknowledgement are provided in www.ebics.org.

2.8.4 Clarification of the Term “Technical Subscriber“

The following examples are to clarify typical actions of the customer server involving a technical subscriber and one or more human subscribers:

PartnerID of the customer	CUSTOM1
UserID (field SystemID) of the technical subscriber	TECHS1
UserID of the user = human subscriber	HUMANS1, HUMANS2, HUMANS3

case 1: The (human) subscriber initiates the download of account statements (STA) via a technical subscriber		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	HUMANS1	STA

SystemID	TECHS1	

case 2: The technical subscriber is due to download account statements (e.g. via order type STA) automatically, e.g. overnight		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	TECHS1	STA
SystemID	TECHS1	

case 3: HVU overview is based on (human) users (e.g. HUMANS1) because only (human) subscribers may possess and provide bank-technical ES's which are essential to complete orders.		
Content of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	HUMANS1	HVU
SystemID	TECHT1	

case 4: The (human) subscriber initiates the upload of a payment file (e.g. via order type IZV) and subscribes it by his own (if further signatures are necessary for processing of the order, it is stored intermediately in the Distributed Electronic Signature (VEU)).		
Content of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	HUMANS1	IZV
SystemID	TECHS1	
ES provided by	HUMANS1 (T/E/A/B)	IZV

case 5: The (human) subscriber initiates the upload of a payment file (e.g. via order type IZV) which was subscribed by other (human) subsribers (if further signatures are necessary for processing of the order, it is stored intermediately in the Distributed Electronic Signature (VEU)).		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	HUMANS1	IZV
SystemID	TECHS1	
ES provided by	HUMANS2 (T/E/A/B)	IZV
	HUMANS3 (T/E/A/B)	IZV

case 6: The technical subscriber initiates the upload of a payment file (e.g. via order type IZV) which was subscribed by other (human) subsribers (if further signatures are necessary for processing the order, it is stored intermediately in the Distributed Electronic Signature (VEU)).		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	TECHS1	IZV
SystemID	TECHS1	
ES provided by	HUMANS2 (T/E/A/B)	IZV
	HUMANS3 (T/E/A/B)	IZV

case 7: The (human) subscriber HUMANS1 initiates the intermediate storage of an IZV file in the Distributed Electronic Signature (VEU) and subscribes it by his own.		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	HUMANS1	IZV
SystemID	TECHS1	
ES provided by	HUMANS1 (T/A/B)	IZV

case 8: The technical subscriber TECHS1 automatically initiates the intermediate storage of an IZV file in the Distributed Electronic Signature (VEU).		
Contents of the fields in the EBICS request		**Authorisation order type**
PartnerID	CUSTOM1	
UserID	TECHS1	IZV
SystemID	TECHS1	
ES provided by	TECHS1 (T)	IZV

2.8.5 Example for the Interpretation of Field AccountInfo@ID and FileFormat in the Order Types HKD and HTD

To clarify the use of `AccountInfo@ID` an XML example was suitably shortened:

```
<PartnerInfo>
    <AccountInfo>
        <AccountNumber> 111 </AccountNumber>
```

```
        </AccountInfo>
        <AccountInfo>
                <AccountNumber id="accid_2"> 222 </AccountNumber>
        </AccountInfo>
        <AccountInfo>
                <AccountNumber id="accid_3"> 333 </AccountNumber>
        </AccountInfo>
</PartnerInfo>
<UserInfo>
        <UserID> USER_1 </UserID>
        <Permission>
                <OrderTypes>IZG</OrderTypes>
        </Permission>
        <Permission>
                <OrderTypes>IZL</OrderTypes>
                <AccountID>accid_3</AccountID>
        </Permission>
        <Permission>
                <OrderTypes>FUL</OrderTypes>
                <MaxAmount Currency="EUR">6000.00</MaxAmount>
                <FileFormat  CountryCode="FR">pain.xxx.cfonb160.dct</ FileFormat >
        </Permission>
        <Permission>
                <OrderType>FDL</OrderType>
                <FileFormat  CountryCode="FR">camt.xxx.cfonb120.stm</FileFormat >
        </Permission>
</UserInfo>
```

In the example mentioned above the user `USER_1` may use

1. order type IZG in combination with all accounts (`111`, `222` and `333`). Furthermore `USER_1` may use order type IZL only for account `333`.
2. order type FUL in combination with all accounts (`111`, `222` and `333`) but only for the file format pain.xxx.cfonb160.dct and up to the maximum amount (6.000 Euro)
3. order type FDL in combination with all accounts (`111`, `222` and `333`) but only for the file format camt.xxx.cfonb120.stm

3 Different usage of EBICS

France	Germany	Approach for a standardisation
In France the BIC is allocated to the HostID	In Germany the field HostID is generally only 8 characters long	Both allocations are covered by the standard. The different practice of allocation is due to the different initial situations. As the identifiers have been bilaterally agreed on (by each bank and its customer; for the HostID and the UserID and the PartnerID as well), there are no interoperability problems. Note that the HostID is an identifier for a technical system (EBICS bank server) and no bank ID, so it is not necessary to use the BIC for this. It is also possible that banks have several HostID for their (several) EBICS servers.
Only two order types are applied (for upload and download) - The format identifiers are assigned by an EBICS parameter in the order types FUL and FDL.	A multitude of order types - Format is indicated by an order type identifier	The different practice of allocation is due to the different initial situations.The harmonisation has been started simultaneously from both sides: 1) The two „neutral“ order types FUL and FDL are applicable in Germany already today and can be agreed on bilaterally with the customer like any other order type. It is up to the customer’s and the bank’s intention to use these order types, too. Especially, the file format parameter list has been adopted in the Annex 2 of the EBICS specification. Now then Annex 2 lists both order type codes and file format parameters. The file format parameters can also be applied by German banks however these formats are national (French) formats.in the first instance. 2) Moreover the order type codes in annex 2 of the EBICS specification are - at the first instance - at the German market’s disposal. At present, however, especially national (German) formats are listed. The advancing unification of the SEPA customer-bank formats (use of the EPC Implementation Guidelines in France as well in Germany) and the use of additional ISO 20022 formats (e.g. camt) is, however, a starting point for defining common/uniform order types that can be used in both countries. A close arrangement within the EBICS Working Group regarding the continuation of annex 2 (flexible and independent from the EBICS version) is already decided by the Board of Directors (BoD) of the EBICS SCRL. Challenge / to be analized: Common order type list for ISO 20022 formats (for those which are used similarly in France and in Germany).
Authorisation with accompanying notes that are signed by hand (Telefax);	Electronic signature for authorisation and transport	Customers in France already authorise with electronic signature (former ETEBAC5 users). These using EBICS for transport only (former ETEBAC3 users), however, are also behaving in compliance with the standards. Even in Germany, it is practice (although it is not that widespread) to transfer a

France	Germany	Approach for a standardisation
Electronic signature only for transport		file via EBICS and to submit the authorisation separately by way of an accompanying notes that are signed by hand. Marketing for the conversion of customers who, up to now, have been using EBICS only for transport to the electronic signature. Target in France: Conversion of all clients to electronic signatures for authorisation in the next years.
For transfer of public keys X.509 format is mandatory	Use of a proprietory public key format; In addition X.509 format as options (but still not used at the moment)	Also in France, bank and customer are partially exchanging bilateral keys (self signing certificates, generated by the banks and customers respectively). In this case, the process does not differ from the one in Germany where for each customer-bank-relationship key pairs are exchanged also. As certificates issued by a CA are used (at least for the ES key) the letters for INI and HIA can be avoided (using H3K for initialisation). As in Germany the use of X.509 structures is also possible, the starting point is advantageous. The objective in France are consistently accepted certificates by an interoperability policy (certain requirements which are adhered to by all who issue certificates for the authorisation in EBICS). **Recommendations:** 1) The German financial institutions are still at the beginning of using x.509 certificates in EBICS. The coordination of the interoperability policy in the EBICS Working Group ensures an equal/uniform usage of X.509 structures. The mandatory migration from the proprietary key format to the X.509 format is recommended for German banks. For this a “business decision” of ZKA is needed. 2) The French members of the EBICS Working Group regularly report on their experiences with the use of certificates (Standard TOP). 3) an important common objective is to achieve the multiacceptance of certificates, for the key management and later on for the signature process (INI letter can be avoided).
The Distributed Electronic Signature (VEU) is still not applied	To banks are obligated to enable use of VEU to their customers. Otherwise the customer is not obligated to use it.	An analysis of the opportunities of the application of the VEU in France is planned after the two-phase migration of the ETEBAC users to EBICS. A precise date is not fixed yet, but it is the declared intention of the French community to advance quickly in this matter. Support by the German members of the EBICS Working Group by way of an exchange of experiences and a transfer of know-how is ensured.
Specific French Implementation Guide (IG) required for		Contrary to Germany where the basic concept of the communication standards have been maintained by the conversion of BCS/FTAM (previous standard) to EBICS, the migration in France was or is more complex. Thus, an

France	Germany	Approach for a standardisation
development		additional implememtation document was necessary for the first time. With EBICS version 2.5 and the introduction of the common IG (parallel to EBICS 2.5) the French IG has been made redundant from EBICS version 2.5 on.
Use of the PSR (pain.002.ack, ISO 20022) for logging the data transfer.	Up to now proprietary customer protocol PTK for logging the data transfer, the ES verification and additional validations	A common XML based customer acknowledgement has been specified (refer to chapter 10 of the EBICS specification). For reasons of compatibility, the proprietary PTK has to remain an option for the German market for a transition period (specification in the annex of this common IG). In France this specification replaces the previous description of the PSR/ACK.

4 Annexes

4.1 Allocation of the X.509 Structure

The following chapters describe the current specification of certificates in France.

4.1.1 Structure of the certificates for EBICS customer workstations

The certificate for EBICS customers can be self-signed or imported on the customer workstation if it is issued by a private Certificate Authority CA.
Three usages are defined for EBICS and three certificates are required.

Self-Signed Certificate Usage :

(Self-Signed) Certificate for signature (EBICS T only; EBICS T = use of order attributes "DZHNN exclusively)		
Field X509	**Value N/A=Not Applicable**	**Mandatory Y=Yes N=No**
version	=2 (for X509V3)	Y
serial Number	Random Number of maximum 20 Bytes if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[3]	Y
subject (object or DN)	The attribute is « commonname »	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length – rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of the current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	NonRepudiation	Y
ExtendedKeyUsage		N
CRLDistributionPoints	N/A	N

[3] This is valid only for self-signed certificates. The term of validity of CA certificates will depend on the Policy of the CA for this type of certificate

(Self-Signed) Certificate for Authentication **(EBICS T or TS ;** **EBICS TS = use of order attributes "OZHNN" exclusively; the file contains both order data and signature(s))**		
Field X509	**Value**	**Mandatory** **Y=Yes N=No**
version	=2	Y
serialNumber	Random Number of maximum 35 octets if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[3]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length – rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of the current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	DigitalSignature	Y
ExtendedKeyUsage	N/A	N
CRLDistributionPoints	N/A	N
(Self-Signed) Certificate for Encryption (EBICS T or TS)		
Field X509	**Value** **N/A=Not Applicable**	**Mandatory** **Y=Yes N=No**
version	=2	Y
serialNumber	Random Number of maximum 35 octets if self-signed	Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer	=subject	Y
validity	Validity : 5 years[3]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=SubjectKeyIdentifier of the CA or of current certificate	Y
SubjectKeyIdentifier		Y
KeyUsage	keyEncipherment or keyAgreement	Y

ExtendedKeyUsage	N/A	N
CRLDistributionPoints	N/A	N

CA certificate use :

Each bank determines the certificates, compliant with the structure described below, that it agrees for the personal signature.

CA Signature Certificate (Mandatory on hardware device for TS profile)

Field X509	**Value** **N/A=Not Applicable**	**Mandatory** **Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber	Single by AC registered with max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years.	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	NonRepudiation bit or ContentCommitment bit must be set to 1.	Y
ExtendedKeyUsage	id-kp-emailProtection	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	May be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service.	Y but non critical with OCSP
QCStatement	If the qualified certificate contents OID pointing out the certificate is qualified and the private key of the certificate is stored within a SSCD.	Y if Qualified Certificate

CA Authenticate Certificate (On hardware or software device)

Field X509	**Value** **N/A=Not Applicable**	**Mandatory** **Y=Yes N=No**
version	=2 (for X509V3)	Y

serialNumber	Single for AC Name max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	DigitalSignature bit must be set to 1.	Y
ExtendedKeyUsage	id-kp-clientAuth	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	May be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service.	Y but non critical with OCSP

CA Authenticate Encipherment (On hardware or software device)

Field X509	**Value** **N/A=Not Applicable**	**Mandatory** **Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber	Single for AC Name max length 20 Bytes	Y
Signature Algorithm	RSA-SHA2 (256) or SHA1 (160) intermediary phase for 3 years	Y
issuer	=AC DN	Y
validity	3 years	Y
subject (objet ou DN)	User Id including the « CommonName »	Y
subjectPublicKeyInfo	RSA Key with 2048 bits Key Length-rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier	=AC SubjectKeyIdentifier	Y
SubjectKeyIdentifier		Y
KeyUsage	KeyEncipherment bit must be set to 1.	Y
ExtendedKeyUsage	id-kp-emailProtection	N
Subject Alternative Name	(may include mail address) Be careful to critical character	N but non critical if present
Issuer Alternative Name	Be careful to critical character	N but non critical if present
CRLDistributionPoints	Possibly may be completed with AuthorityInformation access if OCSP service.	Y
Freshest CRL	If DeltaCRL is used	Y but non critical with DeltaCRL
Authority Information Access	If OCSP service	Y but non critical with OCSP

4.1.2 Structure of the certificates for EBICS bank servers

For each usage (in the current version 2.5 for authentication and encryption) a certificate is needed. Both server certificates are treated as SSL TLS certificates and must therefore have both the KeyUsage of DigitalSignature and of KeyEncipherment.
The signature certificate is not provided in EBICS 2.5.

Server Certificate for Authentication		
Field X509	**Value**	**Mandatory Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber		Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer		Y
validity	Validity : 5 years[4]	Y
subject (object or DN)	The attribute is « commonname	Y
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier		Y
KeyUsage	DigitalSignature;KeyEncipherment	Y
CertificatePolicies		Y
CRLDistributionPoints		Y
FreshestCRL		N
ExtendedKeyUsage		N

[4] This is valid only for self-signed certificates. The term of validity of CA certificates will depend on the Policy of the CA for this type of certificate.

Serveur Certificate for Encipherment.		
field X509	**Value**	**Mandatory Y=Yes N=No**
version	=2 (for X509V3)	Y
serialNumber		Y
Signature Algorithm	RSA-SHA2 (256)	Y
issuer		Y
validity	Validity : 5 years[4]	Y
		Y
subject (objet ou DN)	The attribute is « commonname	
subjectPublicKeyInfo	RSA key of 2048 bit-length - rsaEncryption	Y
extensions :		
AuthorityKeyIdentifier		Y
KeyUsage	DigitalSignature;keyEncipherment	Y
CertificatePolicies		Y
CRLDistributionPoints		Y
FreshestCRL		N
ExtendedKeyUsage		N

4.2 Customer acknowledgement “PTK” (previous version in Germany)

Customer protocols document the following processes in connection with customer orders:

- Transmission of order data to and from the bank system
- Transmission of ES’s relating to existing orders to the bank system
- Post-processing of orders, insofar as this relates to signature verification, displaying order data or errors in decompression.

Transmission of the order data from chapter 13 of the EBICS specification and the document “EBICS Annex 2 Order Types” takes place in a file-based manner. The message and error texts are defined correspondingly: “Transmit file to bank”, “File downloaded from bank”, etc. For compatibility reasons, these texts are used again in the EBICS context, even where “Transmit data to bank”, “Data downloaded from bank” would be more suitable.

The new order types defined for EBICS necessitate extension of the stipulations regarding content in the “DFÜ-Abkommen” to include customer protocols. This especially relates to protocolling VEU processes. Sub-sections 4.2.3 and 4.2.4 describe how protocols are kept for the corresponding orders. All stipulations for the customer protocol for SEPA data formats are described in chapter 4.2.2.

4.2.1 Customer protocol - stipulations regarding contents and form

The customer protocol is to be created by the bank in accordance with the following stipulations: The following fundamental provisions apply:

- A maximum of 72 characters may be displayed in a line.
- There will be no protocol entry for the post-processing. (Exceptions: ES verification, decompression error, display of file contents)
- The file display (see chapter 4.2.1.2.3 and 4.2.2) will also be displayed in the case of files without ES[5].

[5]Does not apply to unstructured files

4.2.1.1 Stipulations regarding contents

Order type of the customer protocol:
The order type of the customer protocol is PTK.
Storage and retrieval are not part of this specification. The transmission has already been defined in other chapters. Therefore, only form and content are stipulated here.

List of the individual data fields for each action at the bank's end:
The following data is to be documented in the customer protocol for each action at the bank's end:

Data to be documented	Description
Date and time	Date and time of the action on the bank system
Type of action	See Chapter 4.2.6
Host name	EBICS bank system ID (EBICS host ID)
Order type	Clear text on the order type used by the customer to which the respective bank action relates. Example: "Transmit free text file in 7-bit code"; see Appendix (Chapter 13) and document "EBICS Annex 2 Order Types"

If applicable, multiple instances of the following fields (i.e. per user) are present during the ES verification:

- Subscriber ID (`UserID`, see Chapter 12.5 of the EBICS specification)
- Subscriber name (only if available)
- Order number (`OrderID`, see Chapter 12.5 of the EBICS specification)
- Result of the action (see Chapter 4.2.6)

The following entries (with the exception of the file display) are only available during the ES verifications:

Entry	Description
File name on the customer system	"File name of the original file" from ES file; see EBICS Specification (Chapter 14).
File display	Order types (files in DTAUS, DTAZV and SEPA format): Display of the substantive file data[6] corresponding with the contents of the data carrier's accompanying note see Chapter 4.2.1.3. For SEPA see chapter 4.2.2 Other order types[7]: In the case of files with fixed record length, the first and last record are displayed according to the record length specified for each order type. In the case of files with variable record length, the first and last logical record that is defined for the respective operating system is displayed (e.g. the record before the first CR/LF, e.g. the record before the last CR/LF).
Explanatory text in the event of an error	This field is only displayed when the result of the action "ES verification" shows an error. It is to be understood as a sub-field that explains the concrete error situation (if applicable, per user and per logical file); example: "Agreed amount limit exceeded", see Chapter 4.2.6

4.2.1.2 Stipulations regarding form

The formal configuration of the customer protocol is in accordance with the following stipulations:

4.2.1.2.1 Protocolling the actions at the bank's end

Contents	Format	Length	Example	Comments
1st line				
Date	dd.mm.yy	8	14.11.02	
Spaces		1		
Time	hh:mm:ss	8	11:39:05	
Spaces		5		
Type of action		<=50	Transmit file to bank	See Chapter 4.2.6
3rd line				
Spaces		9		
Text: "Auftrag"		7	order	
Spaces		4		
Colon		1	:	This character is

[6] In the case of "Collective ES's" (several logical files with one ES) display takes place for each logical file

[7] In the case of 8-bit files, the first and last record are displayed as HEXDUMP.

				always located at the 21st position.
Spaces		1		
Text of the order type		41	`Transmit domestic payment transaction order`	If necessary, fill with blank spaces
Space		1		
Order type identification		3	`IZV`	See Chapter 13 of the EBICS specification and document “EBICS Annex 2 Order Types”
Spaces		1		
Order number		4		
Other lines				
Spaces		9		
Type of protocol entry		11	`Result`	If necessary, fill with blank spaces
Colon		1	`:`	This character is always located at the 21st position.
Spaces		1		
Text of the respective protocol entry		<=50	`Transmission OK [01]`	

It is documented that the remote data transmission orders have been processed with encryption and compression by appending two additional text lines to the result line.
The first additional line documents the encryption of the remote data transmission order, the second line documents its compression.

1st additional line:

- 22 spaces indentation
- Text: “Encrypted data transmission[04]”

2nd additional line:

- 22 spaces indentation
- Text “Compressed data transmission[05]”

Examples of protocolling as a whole:

```
14.11.02 11:40:05     Datei zur Bank uebertragen
         Hostname   : EBIXHOST
         Auftrag    : Beliebige Datei senden                    FTB AAI0
         Teilnehmer : USER Teilnehmer User
         Ergebnis   : Transmission successful [01]
                      Data transfer encrypted [04]
```

Data transfer compressed [05]

4.2.1.2.2 Protocolling of errors during signature verification

Subscriber-related errors during signature verification:
As subscribers of different customers may sign within the VEU, details on subscriber, bank code, and account number have to be provided in additional lines.

Contents	Length	Example	Comments
1st line			
Spaces	9		
Text “EU von“	6	ES of	
Spaces	1		
User ID	8	USER0001	
Colon	1	:	This character is located at the 25th position.
Spaces	1		
Error text and error number	<=46	Agreed amount limit exceeded [72]	See Chapters 4.2.6 and 4.2.1.2.5.
2 nd line			
Spaces	9		
Text „Teilnehmer :“	12	Teilnehmer :	
Space	1		
Partner-ID	8		Partner-ID
Space	1		
User-ID	8		User ID, assigned to the prementioned Partner ID
Leerzeichen	1		
Name	<=32		Optional: Name in plain text (alphanumeric characters)
3 rd line			
Spaces	9		
Text „Bank-Code :“	12	Bank-Code :	
Space	1		
Bank-Code	<=50		Left-justified: If the error relates to a specific account, then the declaration of the BIC i.e. national bank-code is mandatory
4 th line			
Spaces	9		

Text „Kontonummer :“	12	`Kontonummer:`	
Space	1		
Account number	<=50		Left-justified: If the error relates to a specific account, then the declaration of the IBAN i.e. national account number ist mandatory

Example:

```
EU von E7503480 : No account authorisation [71]
Teilnehmer : E750348Z E7503480 Mustermann, Franz
Bank-Code  : COBADEFFXXX
Kontonummer: DE89370400440532013001
```

General error texts during signature verification:

Contents	Lenght	Example	Comments
General error messages for signature verification			
Spaces	9		
Error text and error code	<=63	`Die erforderliche Anzahl EUs ist nicht vorhanden [33]`	See Chapters 4.2.6 and 4.2.1.2.5.

Example of general error messages for signature verification:

```
Waiting time expired due to incomplete order [55]
```

4.2.1.2.3 File display

Contents	Length	Comments
File display		
Spaces	4	See Chapter 4.2.1.3 for an example File display at customer’s & bank’s end (DTAUS and DTAZV format) and chapter 4.2.2 for SEPA payments
File display	68	

4.2.1.2.4 Inserting individual texts

Bank-individual texts may be inserted in the customer log file. Such texts can e.g. include processing information of the bank’s host or specific customer information. For the PTK log files to be automatically evaluable the information is marked accordingly:

For marking purposes the first line of the individual text always contains the words „ADDITIONAL INFORMATION“ and is inserted like the first line of a PTK log entry marked as „kind of activity“ including time stamp (see chapter 4.2.1.2.1 Protocolling the actions at the bank’s end). The end is marked by the time stamp of the following PTK log entry analog to all PTK log entries.

Example:

26.10.05 11:15:00 ADDITIONAL INFORMATION
==========
THIS IS TO INFORM YOU THAT THE ELECTRONIC SIGNATURE CODE A005/6- .....

4.2.1.2.5 Support of foreign-language customer protocols

The customer protocol can optionally be generated in other languages as well as German. In this connection, it should be noted that information contained in the protocol that is evaluated by machine at the customer’s end after download (e.g. ES verification results) must be marked separately. In this way, it can be ensured that machine evaluation of the protocols generated in the various languages functions in the customer software. To this end, all information that is of importance for machine evaluation is to be marked by the attachment of an unambiguous 2-digit number. The actual text will be separated from the unambiguous number by a space. The number will be contained within brackets “[ ]”. After carrying out a protocol retrieval, the unambiguous numbers can then be correspondingly interpreted by the customer system within the framework of machine evaluation, independent of the language.

Hence the following structure results for the texts in the customer protocol that may be subject to machine evaluation:

TTTX’20’[NN]

- TTT actual text
- X’20 space as separator between text and number
- [NN] 2-digit, bracketed number that must be unambiguous

Machine evaluation is generally carried out on those texts that show the results of the remote data transmission order, including the signature verification. The following table shows a list of the text numbers and the associated texts that may be subject to machine evaluation. For reasons of clarity, the individual texts are divided into the sections “Remote data transmission”, “Electronic signature”, “File-based post-processing” and “Bank-technical verifications”.

Text number	Text
Remote data transmission (section 1-20)	
01	Transmission OK

Text number	Text
02	Transmission cancelled
04	Data transfer encrypted
05	Data transfer compressed
07	No data available
Electronic signature (section 21-50)	
21	Signature verification
22	Original file belonging to ES not yet transmitted
23	Signature(s) not yet transmitted
24	Signature(s) OK
25	Error with signature(s)
26	Subscriber has signed more than once
27	No signature authorisation
28	Signature is incorrect
29	Identical signature found
30	Incorrect public key version
31	No public key available
32	Public key not yet activated
33	The required number of ES's is not present
34	Specifications of original file not identical for al ES's
35	File cannot be verified. Completely repeat the order !
36	Incorrect structure or size of the ES file
37	Insufficient ES authorisation(s)
File-related post-processing (section 51-70)	
51	Decompression error
52	Cannot read file
53	Decryption error
54	File structure error
55	Waiting time expired due to incomplete order
56	Order file deleted
57	Transfer to pass by accompanying note signed by hand
58	Transmission incorrect, Order file deleted
Bank-technical checks (section 71-90)	
71	Not authorised for account
72	Agreed amount limit exceeded

Examples:

```
14.11.02 11:50:15     Datei zur Bank übertragen
         Hostname   : EBIXHOST
         Auftrag    : Send any file                              FTB AAI0
         Teilnehmer : USER subscriber User
         Ergebnis   : Transmission successful [01]
                      Data transfer unencrypted [03]
```

```
                    Data transfer uncompressed [06]
```

```
14.11.02 11:50:15     Datei zur Bank übertragen
         Hostname   : EBIXHOST
         Auftrag    : Send any file                              FTB AAJ0
         Teilnehmer : USER subscriber User
         Ergbenis   : Transmission successful [01]
                      Data transfer encrypted [04]
                      Data transfer compressed [05]
```

```
14.11.02 11:51:55     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : Domestic payment transaction file          IZV AAM0
         Teilnehmer : USER subscriber User
         Ergebnis   : Signature(s) OK [24]

Order display file
```

```
14.11.02 11:51:55     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : Domestic payment transaction file         IZV AAN0
         Teilnehmer : USER subscriber User
         Ergebnis   : Error with signature(s)     [25]

Order display file

         Insufficient numbers of signatures [33]
```

4.2.1.2.6 Protocoling of orders which are not autorised in the EBICS process

Orders can be authorised outside the EBICS process (for example, by a accompanying note signed by hand). In this case, the order attributes of the upload order are set to "DZHNN" for the transmission of a payment order. Within the EBICS transaction an electronic signature of signature class "T" is transmitted along with the payment order to the bank. The order is not passed on to the VEU, but directly to the subsequent bank-specific processing. Orders authorised by accompanying notes signed by hand are recorded in the customer protocol after their submission as follows:

If the transport signature was correct:

```
14.10.07 11:51:55     Datei zur Bank übertragen
Hostname   : EBIXHOST
Auftrag    : (Auftrag mit Auftragsart und –nummer)
Teilnehmer : USER Teilnehmer User
Ergebnis   : Transmission successful [01]
Data transfer encrypted [04]
Data transfer compressed [05]

Order display file

         transfer to pass by accompanying note signed by hand [57]
```

If the transport signature was not correct:

```
14.10.07 11:51:55     Datei zur Bank übertragen
         Hostname   : EBIXHOST
         Auftrag    : (Order with order type and order number)
         Teilnehmer : USER Teilnehmer User
         Ergebnis   : Transmission successful [01]
                      Data transfer encrypted [04]
                      Data transfer compressed [05]

Order display file
         Transmission incorrect, Order file deleted [58]
```

4.2.1.3 File display at the customer's and the bank's end

Order types for files in DTAUS format:

	From field number of the DTAUS specification
Payment type	A3
Bank sort code	A4
Account number	A9
Order party	A6
Date created	A7
Number of payments	E4
Total of all amounts (EUR)	E8
Total of account numbers	E6
Total of bank sort codes	E7
Implementation deadline	A11b

Example

```
    ============================================================
     G U T S C H R I F T E N
     Bank-Code                  : 30040000
     Kontonummer                : 0822511260
     Auftraggeber               : Bank-Verlag
     Erstellungsdatum           : 10.05.00
     Anzahl der Zahlungssaetze  : 1
     Summe der Betraege (EUR)   : 68.672,00
     Summe der Kontonummern     : 00000000001234567
     Summe der Bank-Codes       : 00000000007654321
     Ausfuehrungstermin         : 10.05.2000
```

Order types for files in DTAZV format:

	From field number of the DTAZV specification
Q record information (1 Q record for each logical file)	
Bank sort code	Q3
Customer number	Q4
Order party's data	Q5
Date created	Q6
T record information (1 to n T records for each logical file)	
Order currency	T13
Bank sort code	T3
Account currency	T4a
Account number	T4b
Implementation deadline	T5
Amount	Total of fields T14a and T14b for all T records where the preceding fields T13, T3, T4a, T4b and T5 are identically set. If they are set differently in the same file, this T record information is correspondingly specified more than once.
Z record information (1 Z record for each logical file)	
Number of T data sets	Control total from field Z4
Total of amounts	Control total from field Z3

Example:

```
   ============================================================
    G U T S C H R I F T E N
    Bank-Code                 : 30040000
    Kundennummer              : 0000000001
    Auftraggeberdaten         : KARL MUSTERMANN
                                MUSTERSTR. 1
                                50825 KOELN
    Erstellungsdatum          : 10.05.00
    Auftragswaehrung          : ILS
    Bank-Code                 : 30040000
    Kontowaehrung             : EUR
    Kontonummer               : 1234567890
    Ausfuehrungstermin        : 10.05.00
    Betrag                    : 20.000,000
    Anzahl der Datensaetze T  : 000000000000001
    Summe der Betraege        : 000000000020000
```

4.2.2 Stipulations for protocolling SEPA data formats

Customers may transfer SEPA payments to the bank by means of different variants. There are separate corresponding order types. Depending on the variations, differences arise for the protocolling in the customer protocol. The following variations are supported:

- Specifications for SEPA payment transactions: Submission of a pain message containing one or more PaymentInformation blocks (i.e. one or more ordering accounts and/or dates) as described in chapter 2 of Annex 3 of the “DFÜ Abkommen” (Remote Data Transmission Agreement).
- SEPA Container: Submission of several pain messages each with only one PaymentInformation block in a container

4.2.2.1 Specification for SEPA payment transactions

The customer receives an edited version of the submitted file as a part of the customer protocol. This file contains all relevant information for the identification of the original file.

```
28.01.11 16:29:48     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : SEPA Sammelueberweisung                  CCT WZXD
         Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
         Ergebnis   : Electronic signature(s) correct [24]

    ============================================================
     G U T S C H R I F T E N
     Datei-ID   : 4782647268346
     Datum/Zeit : 28.01.2011/09:30:47
    ------------------------------------------------------------
     Sammlerreferenz            : 46573264784
     Bank-Code                  : WELADEDD
     Kontonummer                : DE78300500000045403327
     Auftraggeberdaten          : XXX
     Anzahl der Zahlungssaetze  : 187
     Summe der Betraege (EUR)   : 68.672,00
     Ausfuehrungstermin         : 28.01.2011
    ------------------------------------------------------------
     Sammlerreferenz            : 46573264783
     Bank-Code                  : WELADEDD
     Kontonummer                : DE78300500000045403327
     Auftraggeberdaten          : XXX
     Anzahl der Zahlungssaetze  : 130
     Summe der Betraege (EUR)   : 44.321,00
     Ausfuehrungstermin         : 01.02.2011
    ------------------------------------------------------------
     Sammlerreferenz            : 46573264782
     Bank-Code                  : WELADEDD
     Kontonummer                : DE93300500000012453678
     Auftraggeberdaten          : XXX
     Anzahl der Zahlungssaetze  : 123
     Summe der Betraege (EUR)   : 12.105,00
     Ausfuehrungstermin         : 28.01.2011
    ============================================================
```

Field “Auftrag” (= Order)

As usual, the field "order" contains the complete text and the order type code of the business transaction. Furthermore, the order number is recorded as the last item. SEPA credit transfers according to the ZKA specification with order type code CCT. SEPA direct debits have to be entered as "SEPA bulk direct debit " with order type code CDD (SEPA Core Direct Debits) resp. CDB (SEPA B2B Direct Debits). There is no difference to the PTK structure of DTAUS/DTAZV.

Definition of other fields (if fields are not mentioned, then there is no difference to the existing PTK structure; thus no further explanations are provided):

Field "Dateiname” (=File-ID):
Contains the ID of the submitted pain message (MessageIdentification).

Field “Datum/Zeit” (=Creation date/time):
Contains the creation date and time of the submitted pain message (CreationDateTime). The data are displayed as follows: DD.MM.YYYY/hh:mm:ss.

Field "Sammlerreferenz":
Display of the payment information identification adopted from the field "PaymentInformationIdentification".

Fields "Bank code" and "Kontonummer” (=Account number):
Fields for the national and international bank or account identification (analogous to DTAUS/DTAZV)

Field “Auftraggeberdaten” (=Name and address of principal):
If this (company) name should extend beyond the end of a line, it will not be wrapped, but truncated as this field has no legal relevance for verification. Furthermore, this measure improves machine readability. Otherwise, there is no difference compared to the previous PTK structure.

Field "Anzahl der Zahlungssätze” (=Number of all transactions):
There is no difference to the previous PTK structure. This value indicates the number of payments per PaymentInformation block and has to be determined as it is optional in the pain message.

Field “Summe aller Beträge (EUR)" (=Total of all amounts (EUR)): There is no difference to the previous PTK structure. This value indicates the sum of all amounts per PaymentInformation block and has to be determined as it is optional in the pain message.

Field "Ausführungstermin” (=Execution date): There is no difference to the previous PTK structure. In case of direct debits, the term "execution date" has to be substituted by the term "due date".

4.2.2.2 SEPA-Container

```
28.01.11 16:29:48     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : SEPA Sammelueberweisung                  CCC WZXD
         Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
         Ergebnis   : Electronic signature(s) correct [24]

    ============================================================
     G U T S C H R I F T E N
     Datei-ID   : 4782647268346
     Datum/Zeit : 28.01.2011/09:30:47
```

```
    ------------------------------------------------------------
     Sammlerreferenz            : 46573264782
     Bank-Code                  : WELADEDD
     Kontonummer                : DE78300500000045403327
     Auftraggeberdaten          : XXX
     Anzahl der Zahlungssaetze  : 187
     Summe der Betraege (EUR)   : 68.672,00
     Ausfuehrungstermin         : 28.01.2011
    ============================================================
     G U T S C H R I F T E N
     Datei-ID   : 4782647268346
     Datum/Zeit : 28.02.2008/09:30:47
    ------------------------------------------------------------
     Sammlerreferenz            : 46573264783
     Bank-Code                  : WELADEDD
     Kontonummer                : DE93300500000012453678
     Auftraggeberdaten          : XXX
     Anzahl der Zahlungssaetze  : 123
     Summe der Betraege (EUR)   : 12.105,00
     Ausfuehrungstermin         : 28.01.2011
    ============================================================
```

If in this case the SEPA payment file has been submitted without a bank-technical electronic signature (i.e. without ES for authorization), the Hash Value for each logical file that is contained in the container is returned in the customer protocol:

```
...
     Ausfuehrungstermin         : 28.01.2011
     Hash-Wert                  : 24 AE 87 34 FE BA 22 12
                                  34 E4 5A 34 54 33 43 23
                                  15 34 55 78 FA F1 33 11
                                  93 67 30 03 19 67 BE FA
    ============================================================
```

Annotation to the field "Hash-Wert" (Hash value) (supplementing chapter 4.2.2.1), all other fields are to be filled as explained in chapter 4.2.2.1:

In case of SEPA orders the 32 byte hash value serves as a backup procedure. It is required for the handling of accompanying notes that are signed by hand in order to determine unambiguously if the accompanying note (placing of orders) has been assigned to the pain message. In case of files which are transferred with an ES, the hash value is to be omitted because data integrity and placing of orders are conducted through the ES. In the customer protocol the hash value is displayed in hexadecimal representation. The single bytes are separated by blank characters in order to improve readability. The hash value displayed will be wrapped after the eighth byte to avoid an unspecific line break. The following bytes of the hash value are displayed in the next three lines, each indented by 24 blank characters, thus placing each of them exactly underneath the first eight bytes.

4.2.3 Protocolling the VEU

Processing of orders of the following VEU order types is not protocolled:

- HVD (retrieve VEU state)
- HVU (download VEU overview)
- HVZ (retrieve VEU overview with additional information)
- HVT (retrieve VEU transaction details)

Only orders of the following types are protocolled:

- HVE (add VEU signature)
- HVS (VEU cancellation)

Within the framework of VEU, EBICS provides signatures from more than one customer for an order. In order that the order number is unambiguously assigned to a customer, a new line “customer” is used in the protocol entries for VEU, containing the customer ID of the initiating party in question. In this way, submission of a signature from more than one customer or signature verification within the framework of the VEU from more than one customer can be documented in the protocol files of all involved customers.

In EBICS, bank-technical upload orders are fundamentally submitted with at least one ES: This can be a transport signature of one or more bank-technical ES’s. If the first ES verification of an order is sufficient for its processing or rejection, the protocolling of this signature verification takes place in accordance with Chapter 4.2.1: In this case, a protocol entry is generated for the action “Signature verification” that also contains the file display of the signed order data.

Example:

```
28.02.05 16:29:48     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
         Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
         Ergebnis   : Electronic signature(s) correct [24]

    ============================================================
     G U T S C H R I F T E N
     Bank-Code                 : 70050000
     Kontonummer               : 0045403327
     Auftraggeberdaten         : XXX
     Erstellungsdatum          : 10.02.05
     Anzahl der Zahlungssaetze : 1
     Summe der Betraege (EUR)  : 2,00
     Summe der Kontonummern    : 222222222
     Summe der Bank-Codes      : 222222222
     Ausfuehrungstermin        : 28.02.2005
    ============================================================
```

However, a protocol entry for the action “Forwarding to VEU” is firstly generated if the first successfully-verified ES is not sufficient for processing of the order.

Example:

```
28.02.05 16:29:48     Forwarding to VEU [38]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
         Ergebnis   : Transfer order [46]
                      Processing OK [47]
```

In this case, the first and each subsequent signature verification will additionally be protocolled as the action “Signature verification for VEU”. The structure of the protocol entry “Signature verification for VEU” is comparable with that of the protocol entry for “Signature verification” in Chapter 4.2.1. The difference lies in the missing file display and in the additional customer’s specification of the initiating party.

Example of successful signature verification:

```
28.02.05 16:37:57     VEU signature verification [39]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
         Kunde      : Kunde111
         Teilnehmer : Kunde111 TLN11000 Name_TLN11000
         Ergebnis   : Electronic signature(s) correct [24]
```

Example of signature verification with errors:

```
28.02.05 16:37:57     VEU signature verification [39]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
         Kunde      : Kunde111
         Teilnehmer : Kunde111  TLN11000  Name_TLN11000
         Ergebnis   : Electronic signature(s) incorrect [25]
         EU von TLN11000: Unterschrift ist falsch
```

The protocol entry “Signature verification for VEU” is used irrespectively of whether the order’s ES was transmitted via the new order type HVE or not. Transmission of an ES via HVE is documented via a protocol entry for the action “Transmit file to bank”. Here, “Add VEU signature” is used as an order text for order type HVE. For the order number, not only the order number of the HVE order is protocolled but also the order number of the order that is signed via HVE. This is included in the customer protocol under "reference".

Example:

```
28.02.05 16:29:48     Datei zur Bank uebertragen
         Hostname   : EBIXHOST
         Auftrag    : VEU-Unterschrift hinzufuegen              HVE A233
```

```
        Referenz   : Inlandszahlungsverkehrsdatei              IZV WZXD
        Kunde      : Kunde111
        Teilnehmer : Kunde222  TLN22000  Name_TLN22000
        Ergebnis   : Transmission successful [01]
                     Data transfer encrypted [04]
                     Data transfer compressed [05]
```

After sufficient bank-technical ES's have been delivered and successfully verified for an order, a further protocol entry is generated for the action "End of signature verification for VEU". This lists the subscribers that have signed the order via bank-technical ES, and documents the result "Forward order for post-processing". In addition, it contains the file display of the signed order data in accordance with Chapter 4.2.1.3.

Example:

```
28.02.05 16:37:57     End of VEU signature verification [40]
        Hostname   : EBIXHOST
        Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
        Kunde      : KUNDE111
        Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
        Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
        Ergebnis   : Order forwarded for post-processing [45]

    ============================================================
     G U T S C H R I F T E N
     Bank-Code                  : 30040000
     Kontonummer                : 0825112600
     Auftraggeberdaten          : XXX
     Erstellungsdatum           : 28.02.05
     Anzahl der Zahlungssaetze  : 1
     Summe der Betraege (EUR)   : 10.000,00
     Summe der Kontonummern     : 222222222
     Summe der Bank-Codes       : 222222222
     Ausfuehrungstermin         : 28.02.2005
    ============================================================
```

Transmission of the cancellation of an order via HVS is protocolled in an analogous manner to transmission of a bank-technical ES via HVE. Here, "VEU cancellation" is used as an order text for order type HVS. For the order number, not only the order number of the HVS order is protocolled but also the order number of the order that is signed via HVS. This is included in the customer protocol under "Referenz".

Example:

```
28.02.05 16:29:48     Datei zur Bank uebertragen
        Hostname   : EBIXHOST
        Auftrag    : VEU-Storno                                HVS A234
        Referenz   : Inlandszahlungsverkehrsdatei              IZV WZXD
        Kunde      : KUNDE111
        Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
        Ergebnis   : Transmission successful [01]
                     Data transfer encrypted [04]
```

```
                    Data transfer compressed [05]
```

The cancellation process is documented with a new protocol entry for the action “Cancel VEU order”. Again the order number of the order that is to be cancelled is used as the order number and not the order number of the HVS order itself.

Example:

```
01.03.05 09:29:56     Cancellation of VEU order [41]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              HVS A234
         Referenz   : Inlandszahlungsverkehrsdatei              IZV WZXD
         Kunde      : KUNDE111
         Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
         Ergebnis   : Order cancelled [42]
```

In the event of successful cancellation of the order (result: “order cancelled”) a final protocol entry is generated for this order. This lists both the subscribers that have approved the order via bank-technical ES and also the subscriber that cancelled the order. At the same time, the final protocol entry documents the result “Order cancelled” and contains the file display of the cancelled order (see Chapter 4.2.1.2).

Example:

```
28.02.05 16:37:57     End of VEU signature verification [40]
         Hostname   : EBIXHOST
         Auftrag    : Inlandszahlungsverkehrsdatei              IZV WZXD
         Kunde      : KUNDE111
         Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
         Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
         Ergebnis   : Order cancelled [42]

    ============================================================
     G U T S C H R I F T E N
     Bank-Code                  : 30040000
     Kontonummer                : 0825112600
     Auftraggeberdaten          : BANK-Verlag
     Erstellungsdatum           : 28.02.2005
     Anzahl der Zahlungssaetze  : 1
     Summe der Betraege (EUR)   : 10.000,00
     Summe der Kontonummern     : 22222222
     Summe der Bank-Codes       : 22222222
     Ausfuehrungstermin         : 28.02.2005
    ============================================================
```

Example for data cleansing (deletion of files that were neither cancelled nor released after a period agreed upon by customer and bank). The specification of subscriber identifications is dispensed with intentionally because several subscriber IDs may be considered:

```
23.07.07 16:29:48     Waiting time expired due to incomplete order [55]
         Hostname   : EBIXHOST
```

```
        Auftrag    : Inlandszahlungsverkehrsdatei IZV WZXD
        Kunde      : KUNDE111
        Ergebnis   : Order file deleted [56]

    ============================================================
     G U T S C H R I F T E N
     Bank-Code                 : 11100000
     Kontonummer               : 100111
     Auftraggeber              : A1
     Erstellungsdatum          : 23.07.07
     Anzahl der Zahlungssaetze : 1
     Summe der Betraege (EUR)  : 111,00
     Summe der Kontonummern    : 100111
     Summe der Bank-Codes      : 11100000
     Ausfuehrungstermin        : 23.07.2007
    ============================================================
```

4.2.4 Protocolling key management

For key management orders, protocol entries for the action “Transmit file to bank” or “File downloaded from bank” are fundamentally generated to document the successful or terminated transmission of order data. In the 3rd line as “Text of the order type”, the protocol entries use the short descriptions in brackets from the following list of key management order types:

- INI (Initial transmit public key)
- PUB (Transmit public key)
- HIA (Initial transmit public key)
- HSA (Initial transmit public key)
- HCA (Transmit public key)
- HCS (Transmit public key)
- HPB (Download bank’s public keys).

Examples:

```
19.05.05 10:07:07     Datei zur Bank uebertragen
         Hostname   : EBIXHOST
         Auftrag    : Initiales Senden Public-Key              INI A0DH
         Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
         Ergebnis   : Transmission successful [01]
                      Data transfer unencrypted [03]
                      Data transfer uncompressed [05]
```

```
19.05.05 10:07:07     Datei von Bank abholen
         Hostname   : EBIXHOST
         Auftrag    : Abholen Public-Keys der Bank             HPB
```

```
        Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
        Ergebnis   : Transmission successful [01]
                     Data transfer encrypted [04]
                     Data transfer compressed [05]
```

PUB, HCA, HCS, and HSA orders require precisely one ES (any signature class) from the subscriber whose key is to be changed or transmitted. Protocolling of the signature verification takes place in an analogous manner to the signature verification of bank-technical upload orders but without the order data being displayed.

Examples:

```
28.02.05 16:29:48     Signature verification [21]
        Hostname   : EBIXHOST
        Auftrag    : Senden Public Key                          PUB A0DK
        Teilnehmer : KUNDE111  TLN11000  Name_TLN11000
        Ergebnis   : Electronic signature(s) correct [24]
```

```
28.02.05 16:29:48     Signature verification [21]
        Hostname   : EBIXHOST
        Auftrag    : Senden Public Key                          HCA A0DL
        Teilnehmer : KUNDE111 TLN11000 Name_TLN11000
        Ergebnis   : Electronic signature(s) incorrect [25]
        EU von TLN11000 : Unterschrift ist falsch
```

4.2.5 Protocolling other system-related orders

Orders of the following types are protocolled by means of simple download protocol entries, in particular without protocolling the contents of the download data:

- HAA (download retrievable order types)
- HKD (download customer and subscriber data)
- HPD (download bank parameter)
- HTD (download customer and subscriber data).

Here, the text in brackets is used as “text of the order type” in the third line of the protocol entries.

Example:

```
19.05.05 10:07:07     Datei von Bank abholen
        Hostname   : EBIXHOST
        Auftrag    : Kunden- und Teilnehmerdaten abholen         HPD
        Teilnehmer : KUNDE222 TLN22000 Name_TLN22000
        Ergebnis   : Transmission successful [01]
```

```
                    Data transfer encrypted [04]
                    Data transfer compressed [05]
```

Example for protocolling the suspension of a key (upload and subsequent ES verification):

```
19.12.07 11:46:59     Unterschrift zur Bank uebertragen
         Hostname   : EBIXHOST
         Auftrag    : Sperrung der Zugangsberechtigung           SPR AF0Z
         Teilnehmer : T997100A  Name_TLNT997100A
         Ergebnis   : Transmission successful [01]
                      Data transfer encrypted [04]
                      Data transfer compressed [05]
```

```
20.12.07 10:07:07     Signature verification [21]
         Hostname   : EBIXHOST
         Auftrag    : Sperrung der Zugangsberechtigung           SPR AF0Z
         Teilnehmer : K9971000  T997100A  Name_TLNT997100A
         Ergebnis   : Electronic signature(s) correct [24]
```

4.2.6 Report texts

The following tables represent a complete overview of all report texts that are possible in EBICS.

Type of action	Report or error report texts (German)	Report or error report texts (English)
Transmission	Datei zur Bank uebertragen Datei von Bank abgeholt Unterschrift zur Bank uebertragen	File submitted to the bank File downloaded from the bank Electronic signature submitted to the bank
Post-processing	Unterschriftspruefung [21] Weitergabe zur VEU [38] Unterschriftspruefung zur VEU [39] Abschluss Unterschriftspruefung VEU [40] Stornierung VEU Auftrag [41] Fehler bei Dekomprimierung [51] Fehler bei Entschluesselung [53] Anzeige Dateiinhalt	Signature verification [21] Forwarding to VEU [38] VEU signature verification [39] End of VEU signature verification [40] Cancellation of VEU order [41] Decompression error [51] Decryption error [53] Display of the file content

Result of action	Report or error report texts (German)	Report or error report texts (English)
Transmission	Uebertragung in Ordnung [01]	Transmission successful [01]
	Abbruch der Uebertragung [02]	Transmisson aborted [02]
	Datenuebertragung verschluesselt [04]	Data transfer encrypted [04]
	Datenuebertragung komprimiert [05]	Data transfer compressed [05]
	Keine Daten vorhanden [07]	No data available [07]
Post-processing	Originaldatei zur EU noch nicht uebertragen [22]	Corresponding original file still not sent [22]
	Unterschrift(en) noch nicht uebertragen [23]	Electronic Signature(s) still not sent [23]
	Unterschrift(en) in Ordnung [24]	Electronic signature(s) correct [24]
	Unterschrift(en) fehlerhaft [25]	Electronic signature(s) incorrect [25]
	Auftrag storniert [42]	Order cancelled [42]
	Auftrag nicht storniert [43]	Order not cancelled [43]
	Auftrag zurueckgewiesen [44]	Order rejected [44]
	Auftrag zur Verarbeitung weitergegeben [45]	Order forwarded for post-processing [45]
	Auftrag uebergeben [46]	Transfer order [46]
	Bearbeitung in Ordnung [47]	Processing OK [47]
	Fehler bei Dekomprimierung [51]	Decompression error [51]
	Datei nicht lesbar [52] *(nur bei Aktion „Anzeige Dateiinhalt“)*	File cannot be read [52] *(only in the case of action “Display file content“)*
	Fehler bei Entschluesselung [53]	Decryption error [53]
	Datei ist in ihrem Aufbau fehlerhaft [54]	Incorrect file structure [54]
	Wartezeit unvollstaendiger Auftrag abgelaufen [55]	Waiting time expired due to incomplete order [55]
	Auftrag geloescht [56]	Order file deleted [56]
	OK *(nur bei Aktion „Anzeige Dateiinhalt“)*	OK *(only in the case of action “Display file content“)*

Explanatory text in the event of ES verification errors	Report or error report texts (German)	Report or error report texts (English)
Texts relating to subscriber	Teilnehmer hat mehrfach unterschrieben [26]	User signed multiple times [26]
	Vereinbarter Hoechstbetrag ueberschritten [72]	limit exceeded [72]
	Keine Unterschriftsberechtigung [27]	No authorisation rights [27]
	Teilnehmer hat sich noch nicht initialisiert	User not yet initialised
	Teilnehmer noch nicht freigeschaltet	User not yet activated
	Teilnehmer gesperrt	User is locked
	Teilnehmereintrag nicht vorhanden	User does not exist
	Unterschrift ist falsch [28]	Electronic signature incorrect [28]
	Identische Unterschrift gefunden [29]	Identical signature found [29]
	Falsche Public Key-Version [30][8]	Public key version incorrect [30][8]
	Kein Public Key vorhanden [31]	Public key does not exist [31]
	Public Key noch nicht freigegeben [32]	Public key not yet activated [32]
	Keine Berechtigung fuer Konto [71]	No account authorisation [71]
General texts	Erforderliche EU-Anzahl nicht vorhanden [33]	Insufficient numbers of signatures [33]
	Angaben zum Auftrag nicht je EU identisch [34]	Different order data in signatures [34]
	Datei nicht pruefbar. Auftrag wiederholen [35][9]	File not testable. Repeat complete order [35][9]
	Aufbau bzw. Groesse der EU-Datei falsch [36]	Wrong structure or size of signatures [36]
	EU-Berechtigung(en) nicht ausreichend [37]	Electronic signature(s) rights insufficient [37]
	Weitergereicht zur Freigabe mittels Begleitzettel [57]	transfer to pass by accompanying note signed by hand [57]
	Auftragseinreichung fehlerhaft, Auftrag geloescht [58]	Transmission incorrect, Order file deleted [58]

[8] This report is protocolled when a customer sends signature files to the financial institution after conversion from an older program version (old ES format) to a new program version (new ES format) without having carried out re-initialisation with regard to a public key change.

[9] This report is displayed in the event of a malfunction during the signature check, e.g. not enough storage space