TECHNICAL REPORT

ISO/TR 14904

First edition
1997-05-15

Road transport and traffic telematics — Automatic fee collection (AFC) — Interface specification for clearing between operators

Transport routier et télématique du trafic — Prélèvement automatique de frais (AFC) — Spécification d'interface pour clarification entre opérateurs

Reference number
ISO/TR 14904:1997(E)

Foreword

ISO (the International Organisation for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental or non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The main task of technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/TR 14904, which is a Technical Report of type 2, was prepared by the European Committee for Standardization (CEN) in collaboration with ISO Technical Committee ISO/TC 204, *Transport information and control systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet central@iso.ch
X.400 c=ch; a=400net; p=iso; o=isocs; s=central

Printed in Switzerland

This document is being issued in the Technical Report (type 2) series of publications (according to subclause G.3.2.2 of part 1 of the ISO/IEC Directives, 1995) as a "prospective standard for provisional application" in the field of *transport information and control systems* because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an "International Standard". It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO/TC 204 Secretariat.

A review of this Technical Report (type 2) will be carried out not later than three years after its publication with the options of: extention for another three years; conversion into a International Standard; or withdrawal.

Annexes A and ZA form an integral part of this Technical Report. Annexes B to E are for information only.

Contents

1. Foreword

The text of ENV ISO 14904:1997 has been prepared by Technical Committee CEN/TC 278 "Road transport and traffic telematics", the secretariat of which is held by NNI, in collaboration with Technical Committee ISO/TC 204 "Transport information and control systems".

This Prestandard is prepared under the authority of CEN/TC 278 "Road transport and traffic telematics" by a project team (PT1) set up by CEN/TC 278 and consisting of a group of experts from the financial and transport sectors.

The Prestandard references standards made by other project teams and working groups under CEN/TC 224, ISO/TC 204 as well as existing standards made by ISO.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom.

NOTE: Normative references to International Standards are listed in annex ZA (normative).

2. Introduction

Integration of payment systems concerns the co-ordination and handling of all payment services for traffic and transport applications. This co-ordination involves:

a) the use of a common payment concept for services within or related to road traffic and transport,
b) the enabling of exchange of payment transactions and operational information between different operators involved in public and private transport services and
c) the method of payment itself, i.e. the access to electronic payment means, for the settlement of these acquired services.

In order to enable the integration of payment systems on a higher (e.g. pan-European) level and make clearing between operators possible, the interfaces involved need to be standardised.

Therefore this Prestandard is designed as an interface specification enabling data to be exchanged between different operators and systems adopting a variety of application specifications.

It should be noted that although the data structures defined in the current version of the Prestandard reflect a focus on information transfers for clearing purposes, the interface specification defined herein supports equally well other types of information transfers required within and between payment systems.

All amendments and additions to this Prestandard will be handled by CEN TC 278/WG1 through the use of a database of data objects maintained by the workgroup.

3. Scope

This Prestandard defines the interfaces for clearing between operators and gives a framework of the common message structure and data elements to be used on the interfaces. Its objective is to make the transfer of payment and Automatic Fee Collection (AFC) related data possible both between different payment systems and between different operators such as collection agents, clearing operators, or providers of public and private transport services.

This Prestandard supports:

a) different payment modes (e.g. pre-payment, post-payment),
b) a wide variety of transport and transport related services (tolling, parking, ferry/bridge/tunnel, public transport, payment for route guidance etc.),
c) operator services (co-ordination between collectors of money and charge points etc.),
d) security and privacy.

It is not within the scope of this Prestandard to define administrative procedures and organisational structures. The specification of a higher (e.g. pan-European) level inter-operable payment system is outside the scope of this Prestandard.

Not described within this Prestandard are indirect (external) participants such as authorities, enacting general or special legislation concerning the payment system and other national regulations.

The models presented in this standard are generic. Simple systems (closed systems) can be designed by selecting subsets of the interface framework described herein.

4. Normative references

This Prestandard incorporates by dated or undated reference, provisions from other publications and standards. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications or standards apply to this Prestandard only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

ISO 7812-1	Identification cards - Identification of issuer - Part 1: Numbering scheme
ISO 7812-2	Identification cards - Identification of issuer - Part 2: Application and registration procedures
ISO 7816-5	Identification cards - Integrated circuit cards with contacts - Registration system for applications in IC cards
ISO 8824	Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)
ISO 8825	Information processing systems - Open Systems Interconnection - Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)
ISO 8583	Financial transaction card originated message - Interchange message specifications
ISO 9735	Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT) - Application level syntax rules (amended and reprinted)
ISO 9594	Information technology - The Directory

5. Definitions

For the purpose of this Prestandard the following definitions apply:

5.1. Apportionment

The allocation of money to transport service operators according to the consumption of the services provided, e.g. a bus operator being paid an amount based on the number of a particular type of customer carried.

5.2. Chained Services

A combination of services that result in a discount and/or access rights in one or more of the consumed services. The discount or access rights are usually given to the *User* as a result of having consumed a previous service.

5.3. Clearing

The operation of re-allocating value generated in the payment system(s) between the various operators in a payment system or between payment systems. This operation reflects commercial agreements existing between those parties. An example of such an operation is the exchange of information between *Service Providers* and an *Issuer* which enables the transfer of money from the *Issuer*, collecting the money from the *User*, to the *Service Provider*.

5.4. Clearing Operator

The entity that collects and possibly aggregates transactions from one or more *Service Providers* for delivery to the *Issuer(s)*. The *Clearing Operator* can also handle the *Apportionment* between the *Service Providers*. In the financial world this operator is equivalent to an Acquirer.

5.5. Collection agent

The entity responsible for selling, reloading or delivering the *Payment Means* to the *User* and collecting the payment from the *User*. The *Collection Agent* can also collect user related application specific data from the *User*.

5.6. Contract

This is the expression of an agreement between two or more parties in a payment system or between payment systems. An example of a contract is the specific relationship between a *User* and an *Operator* in a payment system. The contract in this case defines the conditions under which the user may use the services and the amount to be charged.

5.7. (Intersector) Electronic Purse

An application in an Integrated Circuit Card which stores and manipulates electronic value in a secure way and which replaces cash for payments by the *User*.

5.8. Integrated Payment Systems

A common framework of payment methods and information exchange between operators or payment systems that makes transfer of money from one payment system or operator to another possible (*Clearing/Apportionment*).

5.9. Issuer

The entity responsible for the payment system and responsible for issuing the *Payment Means* to the *User*.

5.10. Operator

Generic term for the entities Issuer, Clearing Operator, Collection Agent and Service Provider.

5.11. Payment Means

The expression of a *Contract* between the *User* and the *Issuer* (or via a *Collection Agent*) that allows the *User* to access the services available in the *Payment System*, e.g. an account in a credit card system or an *Electronic Purse*.

5.12. Payment Method
A combination of a Payment Means, a Payment Mode and a Payment Scope.

5.13. Payment Mode
Parameter defining the time dimension in payment by the *User, e.*g. Pre-payment or Post-payment.

5.14. Payment Scope
The application extent of the *Payment Method*, e.g. national transport or inter-sector.

5.15. Payment System
Financial system that includes the complete process of *Issuing,* use of *Payment Means*, *Clearing* and *Settlement* of transactions.

5.16. Service Provider
The entity that accepts the *User's* payment means and in return provides the service to the *User*. Automatic fee collection information created at the point of service consumption is collected by the *Service Provider*.

5.17. Settlement
Transfer of funds from one *Operator* to another according to the *Clearing* rules.

5.18. User
The entity that uses services provided by the *Service Provider* according to the terms of the *Contract* expressed by the *Payment Means.* The *User* receives and reloads the electronic *Payment Means* through the *Collection Agent*.

6. Basic interfaces for clearing between operators

This Prestandard identifies the following minimum set of basic interfaces required for clearing between operators within a payment system and between payment systems:

Operators interfaced	Interfaces covered by the standard	Interfaces NOT covered by the standard
User - Service Provider		X
Service Provider - Clearing Operator	X	
Clearing Operator - Issuer	X	
Issuer - Collection Agent	X	
Collection Agent - User		X
Clearing Operator - Clearing Operator	X	

NOTE: The interface specification defined in this Prestandard is designed to be flexible enough to accommodate any additional operator-to-operator information transfer paths which may be required by the integration and operation of payment systems.

7. Interface framework

7.1. Introduction

This section defines a common message structure to enable the exchange of data on any of the 4 interfaces specified in section 6.

The common message structure is summarised in section 7.2 and described in more detail in the subclauses hereafter.

Message class, message type, sender ID, receiver ID and message ID are only normative requirements when they are not provided by other communication layers.

7.2. Summary of message structure

Figure 7.1 shows graphically the message structure for the Automatic Fee Collection (AFC) related Protocol Data Unit (PDU). The objects shown in the diagram can either be unsecured or secured globally or individually.

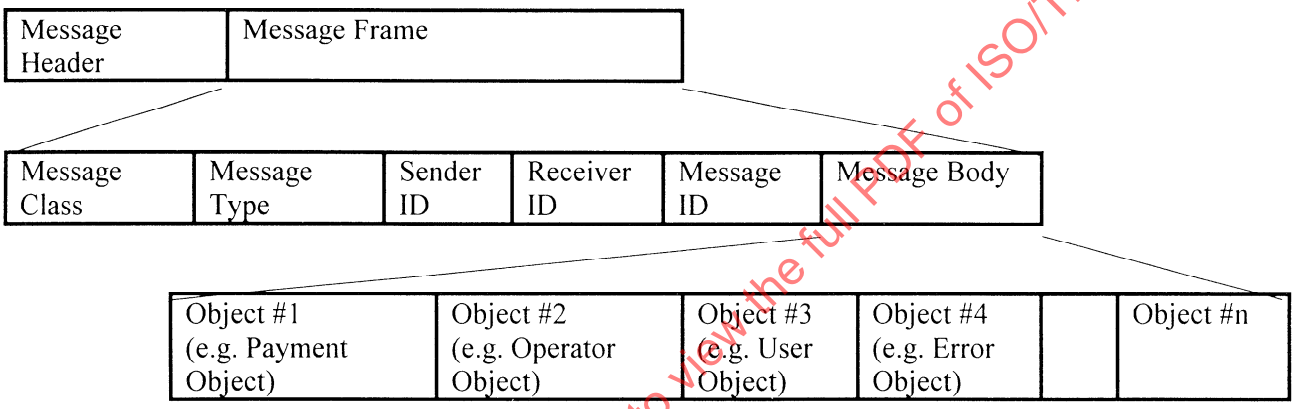

Figure 7.1: Message Structure

7.3. Message Header

At the beginning of each message is a message header. The message header contains a version identifier.

The version identifier is an integer that identifies the version of the protocol. As this integer will always be the first element in the sequence, the receiving party will always be able to identify the version of the protocol being used to send the data. This Prestandard defines version 1 of the protocol.

7.4. Message frame

This part of the message contains the core data called Protocol Data Unit (PDU). Within the description of the message frame there is a choice between a number of different protocol data units. These PDU's can be of the following type:

- AFC related PDU (as described in this Prestandard)
- any other PDU that is defined externally.

NOTE: External PDUs can include: ISO8583 PDU, CEN/TC224 PDU, EDIFACT PDU etc.

The Message Frame for the AFC related PDU contains the following elements:

a) Message Class
b) Message Type
c) Sender ID
d) Receiver ID
e) Message ID
f) Message Body

These elements are described further in the following subclauses.

7.4.1. Message class

The Message Class defines the purpose of a message. All messages belong to one of six different message classes. The different Message Classes defined in this Prestandard are :

a) Request
b) Request Response
c) Advice
d) Advice Response
e) Notification
f) Notification Acknowledgement

7.4.1.1. Request

A Request is a class of message in which the sender requests some action and/or some information from the receiver. The request could for example be a request for a transaction approval or a request for reconciliation totals.

7.4.1.2. Request Response

A Request Response is a class of message that is sent in response to a request message. The response should be sent when the requested action is performed and/or when the requested information is available. The response contains the result of the action or the requested information.

7.4.1.3. Advice

An Advice is a class of message that contains information that is vital for the proper function of the system. This could be, e.g. a status list or key management data.

7.4.1.4. Advice Response

An Advice Response is a class of message that is sent in response to an advice message. The response does not carry any requested data, but is merely a reply that indicates whether the advice was or is going to be followed or not.

7.4.1.5. Notification

A Notification is a class of message in which the sender informs the receiver of some action the sender or some other party has performed. This could for example be information about a report that has been sent to the authorities.

7.4.1.6. Notification Acknowledgement

A Notification Acknowledgement is a class of message that can be sent in response to a notification message.

7.4.2. Message type

The Message Type is the basic selection between different types of messages.

The various data objects which are needed to carry out the payment system functions are sent through the interface grouped according to broad message types. The different Message Types defined in this Prestandard are:

a) Services List
b) Fare Products List
c) Customer Details
d) Apportionment Rules
e) Reconciliation Totals
f) Authorisation
g) Transaction
h) Report Sent
i) Key Management
j) Status List
k) Equipment Status
l) Event Exception
m) Payment Method Acceptance
n) Undefined Message Type (reserved for future)

7.4.2.1. Services List
This message type is used by the sender to send a list of available services.

EXAMPLE: This message type could be used by the Issuer to inform the Collection Agent of the possible services available to the user. The Collection Agent may need such information to create the customers contract, e.g. in parking, public transport and tolling.

7.4.2.2. Fare Products List
This message type is used by the sender to inform the receiver of the different fare products that are available.

EXAMPLE: This message type could be used by the Issuer to inform the Collection Agent of the range of service products that the user can purchase and the price to be charged for these fare products, e.g. a monthly bus ticket.

7.4.2.3. Customer Details
This message type is used to send specific customer/user data.

EXAMPLE 1: This message type can be used to send messages from the Collection Agent to the Issuer to inform the Issuer of the details of the contract with the customer. The message may then include details on the user's age, user profile, etc.

EXAMPLE 2: This message type can also be used by the Issuer to send information to the Clearing Operator that may effect the clearing process. This information could include the amount of discount to be applied to the customer's transaction or the amount to be deducted in the case of chained services.

7.4.2.4. Apportionment Rules
This message type is used to inform the receiver of the message what kind of apportionment rules apply.

EXAMPLE 1: This message type can be used by the Issuer to inform the Collection Agent of the rules that may have to be applied when forwarding payment to the Issuer. Details include the amount of commission that the agent is entitled to and can also include the proportion of payment to be made.

EXAMPLE 2: The message type could also be used to inform the Clearing Operator of the apportionment rules that may be required when operating the clearing process. These may include the discounts to be applied in the case of chained services.

7.4.2.5. Reconciliation Totals
This message type is used to send transaction totals (i.e. not individual transactions) between operators.

EXAMPLE: This message type can be used for requesting totals, e.g. a Clearing Operator requesting totals from a Service Provider. The response on this latter request will then include the totals.

7.4.2.6. Authorisation
This message type is used to obtain on-line authorisation from the Issuer for a particular transaction.

7.4.2.7. Transaction
This message type is used to carry the transaction data.

EXAMPLE: This message type can contain transactions relating to details of sales of a Collection Agent or transactions from a Service Provider to a Clearing Operator. In all situations, the messages can include transport specific details from a transaction.

7.4.2.8. Report Sent
This message type is used by the sender to inform the receiver that a report has been sent to an external authority.

7.4.2.9. Key Management
This message type is used to send keys or other security related information.

EXAMPLE: It may be used by the Issuer to send new public keys to the Clearing Operator or to inform him of a different version of cryptographic algorithm being implemented.

7.4.2.10. Status List
This message type is used to send out information regarding the status of a User, Service Provider or other operator in the system. The class of the message (Request, Advice and Notification) will inform the receiver of the message on what to do with the status list.

EXAMPLE: The objects that are included to be rejected (statuslisted) on the status list need not be only payment means, but a status list from an Issuer to a Clearing Operator may also include black listed Service Providers, i.e. transactions are no longer to be accepted from the Service Provider.

7.4.2.11. Equipment Status
This message type is used by the sender to send and/or request information on equipment status.

EXAMPLE 1: The messages may then include the status of the equipment as active, on test, out of service.

EXAMPLE 2: It may also be used to send certain control totals to the Clearing Operator i.e. transaction counters on the equipment.

7.4.2.12. Event Exception
This message type is used to transfer information related to the procedures taken/to be taken to ensure transaction completion in the event of irregularities or fraud.

EXAMPLE: It may be used to send the data regarding the event to be enforced. This could include a video picture of the event.

7.4.2.13. Payment Method Acceptance
This message type is used to sent out information by the Issuer on rules and supplementary information on acceptance of a specific payment method, e.g. currency conversion, maximum values, required logging etc.

7.4.2.14. Undefined Message Type
This message type is used to indicate that the message type is undefined and reserved for future use.

7.4.3. Identity of the sender
The identity of the sender is defined as being either a simple type or complex type. The simple type of identifier identifies the sender as the type of abstract entity (Issuer, Clearing Operator, Service Provider, Collection Agent). The complex identifier shall be unique within the payment system.

EXAMPLE: The sender ID number can be defined in accordance with X.501 or ISO 7812 or other numbering schemes.

7.4.4. Identity of the receiver
The identity of the receiver is defined as being either a simple type or complex type. The simple type of identifier identifies the receiver as the type of entity (Issuer, Clearing Operator, Service Provider, Collection Agent). The complex identifier shall be unique within the payment system.

EXAMPLE: The receiver ID number can be defined in accordance with X.501 or ISO 7812 or other numbering schemes.

NOTE: The sender and receiver ID do not include or apply to the entity User.

7.4.5. Message identifier
The message identifier, together with the sender and receiver identities, identifies an actual message. The sender determines a unique identifier for the message. If the receiver of the message replies, the reply shall contain the same message identifier.

By using this message identifier a response to a message is always uniquely linked to the original message.

7.4.6. Message Body
The Message Body is a sequence of data objects. These data objects are defined in the following sections of this Prestandard.

7.5. Security data

Part of the information which will be exchanged over the interfaces covered by this Prestandard will constitute an important asset for the respective parties involved. While meeting the security needs of a closed system remains the domain of the parties concerned, an interface specification constitutes a common ground for the implementation of real-world interfaces for clearing between operators within the scope of a higher (e.g. pan-European) level integrated payment system. The interface specification shall make sufficient provision to incorporate current and future security related items.

The security data at the message level and the secured data objects will provide support for security related items. The various security issues can be stated as follows:

Confidentiality	Sensitive data and information are available only to authorised parties (confidentiality of contents). In addition to pure financial transaction information which may naturally be subject to tampering, other, more transport related types of information are to be carried through the same interface (i.e. volumes, type of operations, details of activities, network etc.). This information may prove very sensitive in an increasingly competitive environment.
Integrity	Sensitive data, information and message sequencing are guarded in such a way that any alteration or destruction by unauthorised parties is detected (integrity of contents, integrity of message sequence).
Authentication	The origin and destination of information and the entities involved in the exchange of information are authenticated (message origin authentication, message destination authentication, peer entity authentication).
Non-repudiation	Protection against the denial, by one of the parties involved in the communication through the interface, of having participated in all or part of the communications. Support for the following forms of non-repudiation services may be required: Non-repudiation with proof of origin Non-repudiation with proof of delivery Non-repudiation with proof of submission
Availability	Data, information are available to authorised parties.
Auditing/Accountability	Protection against anomalies in the flow of transactions by the use of time variant parameters. This may also include recording of system activity for security related monitoring purposes.

8. Method of description

The data types used in the interface are specified using Abstract Syntax Notation One (ASN.1). This allows for a flexible, yet unambiguous use of the interface as new data types can be defined that are uniquely recognisable by interfacing parties.

To encode the data types specified in abstract notation into a transmittable data stream, encoding rules are used.

To ensure inter-operability on a higher (e.g. pan-European) level, BER (Basic Encoding Rules) shall be used, unless the two interfacing parties have bilateral agreements which specify the use of other encoding rules.

NOTE 1: ASN.1 (Abstract Syntax Notation One) is a formal language that defines a set of primitive data types and provides a facility to construct new elements with their own typing inherent in the structure. The data types used in the interface are specified using Abstract Syntax Notation One (ASN.1). This notation allows for the definition of abstract syntaxes, enabling application layer standards to define the types of information required to transfer using the presentation service."

NOTE 2: BER (Basic Encoding Rules) is a transfer syntax notation which maps the ASN.1 into a form where each data type is encoded as tag, length and value.

NOTE 3: Since all data types are described using ASN.1, they always will be transmitted in the correct format, irrespective of the encoding rules used. Of the many encoding rules currently defined two types of encoding rules are most common: BER (Basic Encoding Rules) and PER (Packed Encoding Rules).

The description includes the basic data elements that two communicating parties will need. If additional data elements are needed, the description can be extended to include these elements. It is also possible for parties, other than the ones covered by this standard, to engage in a communication using this communication protocol.

9. Message

The Message type describes the complete data transferred between two operators. The data is contained within a sequence.

```
Message ::=
     SEQUENCE  {
          version
                                        INTEGER
                                             { version1(1) },
          data
                                        ProtocolDataUnit
     }
```

To identify different versions of the protocol, the message starts with an integer identifying the version. Since this integer will always be the first element in the sequence, the receiving party will always be able to identify the version of the protocol in use. This Prestandard defines version 1 of the protocol.

The ProtocolDataUnit is a choice between two types of data structures.

```
ProtocolDataUnit ::=
     CHOICE  {
          AFCrelated [0]
                                        AFCRelated-PDU,
          other
                                        EXTERNAL
     }

AFCRelated-PDU
     CHOICE {
          globallySecuredAFCData
                                        EXTERNAL
          locallySecuredAFCData
                                        LocallySecuredAFCData
     }
```

NOTE: Locally Secured AFC Data means that the data object may or may not be associated with a security object. This allows data objects to be secured or unsecured individually.

The locallySecuredAFCData is the data structure defined in this Prestandard.

The EXTERNAL data type allows for data types defined by other standards to be encapsulated in the ProtocolDataUnit used in this Prestandard. This could be used for example when two parties are using an existing standard and want to continue using that standard for some types of data transfers. By encapsulating that standard within the message structure defined in this Prestandard, the same communicating link can be used.

Since the data type is EXTERNAL, no interpretation of the data can be done using this Prestandard. The EXTERNAL data type can include identifiers that identify what kind of data is included. The identities used can either be based on bilateral agreements or defined using global OBJECT IDENTIFIERS. This Prestandard does not define any identifiers for this data type EXTERNAL.

10. AFCRelated-PDU - locallySecuredAFCData

The AFCrelated-PDU introduced in section 7, has a data structure LocallySecuredAFCData with the following definition:

```
LocallySecuredAFCData ::=
     IMPLICIT SEQUENCE  {
          messageClass
               MessageClass                    OPTIONAL,
          messageType
               MessageType                     OPTIONAL,
          senderID
               SenderAndReceiverID             OPTIONAL,
          receiverID
               SenderAndReceiverID             OPTIONAL,
          messageID
               MessageID                       OPTIONAL,
          messageBody
               MessageBody
     }
```

10.1. MessageClass

All messages belong to one of six different message classes. Section 7 of this Prestandard contains the description for the different classes.

```
MessageClass ::=
     ENUMERATED {
          Request (0),
          RequestResponse (1),
          Advice (2),
          AdviceResponse (3),
          Notification (4),
          NotificationAcknowledgment (5)
     }
```

10.2. MessageType

The message type is the basic selection between different types of messages. Section 7 of this Prestandard contains the description for the different types.

```
MessageType ::=
     ENUMERATED  {
          Undefined (0),
          ServicesList (1),
          FareProductsList (2),
          CustomerDetails (3),
          ApportionmentRules (4),
          ReconciliationTotals (5),
          Authorisation (6),
          Transaction (7),
          ReportSent (8),
          KeyManagement (9),
          StatusList (10),
          EquipmentStatus (11),
          EventException (12),
```

```
        PaymentMethodAcceptance (13).
    }
```

10.3. SenderAndReceiverID

The SenderAndReceiverID is used to identify the sender or the receiver. The identity is a choice between four different types.

```
SenderAndReceiverID ::=
    CHOICE  {
        simpleID [0]
                            EntityID,
        cenTC224WG11ID [1]
                            CENTC224WG11-ID  -- import from prEN1545-1,
        x501ID [2]
                            X501-ID,    -- import from ISO9594
        cenTC278WG12ID [3]
                            CENTC278WG12-ID  -- import from prENV12314-2
    }
```

10.3.1. EntityID

The EntityID is an enumerated type identifying one of the entities defined in section 5 of this Prestandard. This type is used when a less complex identity identifier is required.

```
EntityID ::=
    ENUMERATED {
        Undefined (0),
        Issuer (1),
        ClearingOperator (2),
        ServiceProvider (3),
        CollectionAgent (4),
        User (5)
    }
```

Undefined is used when the sender or receiver is not an entity of the conceptual model.

10.3.2. CENTC224WG11-ID

The CENTC224WG11-ID is used when an identity can be referenced through the use of a numeric string based on prEN1545-1.

```
CENTC224WG11-ID ::=
  IMPLICIT NumericString (SIZE(7..19))
```

10.3.3. ISO9594
The X501-ID is used when an identity can be referenced through the numbering schemes based on ISO9594.

10.3.4. CENTC278WG12-ID
The CENTC278WG12-ID is used when an identity can be referenced through the numbering schemes based on prENV12314.

10.4. MessageID
The `MessageID` is an identifier that, together with the sender and receiver identities, identifies an actual message. It is used for message sequencing in simple transfer systems.

```
MessageID ::=
      INTEGER
```

10.5. MessageBody
The `MessageBody` is a container for all of the different secured or unsecured information objects that occur in a message. There can be zero or more objects, which are of type `MessageObject`.

```
MessageBody ::=
     SET OF
          MessageObject
```

`MessageObject` is a sequence of two elements.

```
MessageObject ::=
     SEQUENCE  {
          messageObjectType
               OBJECT IDENTIFIER,
          messageObjectContent
               ANY
                    DEFINED BY messageObjectType
     }
```

`messageObjectType` is an object identifier that uniquely identifies the subsequent object. This could be an identifier that is defined in this Prestandard (see Annex A) or any other unique identifier. This means that the message body can contain both information objects defined in this Prestandard as well as information objects that are defined elsewhere.

The `messageObjectContent` is the information object associated with `messageObjectType`.
In sections 11 to 15 special AFC related information objects are defined.

11. ErrorCode

The `ErrorCode` information object is sent when an error needs to be reported.

```
ErrorCode ::=
  INTEGER
```

12. ReportSent

This information object is used to inform that a report has been sent.

```
ReportSent ::=
     SET  {
          reportNumber [0]
               IMPLICIT INTEGER                    OPTIONAL,
          sentDate     [1]
               IMPLICIT UTCTime                    OPTIONAL,
          sentTo       [2]
               IMPLICIT PrintableString            OPTIONAL,
          reportText   [3]
               IMPLICIT PrintableString            OPTIONAL
     }
```

The interpretation of the different elements is based on agreements between the sender and receiver.

13. StatusList

The StatusList object is used to transmit status information. The status list may specify other types of targets other than payment means, e.g. Service Provider(s) from which transactions are no longer to be accepted.

```
StatusList ::=
     SEQUENCE  {
          type
               StatusListType,
          operation
               StatusListOperation,
          objectList
               SET OF
                    StatusListObject
     }
```

13.1. StatusListType

The StatusListType describes the type of the status list sent. It could either be a Negative or a Positive status list.

```
StatusListType ::=
     ENUMERATED {
          Negative (0),
          Positive (1)
     }
```

A Negative list is a list that contains objects that, possibly under certain constraints, are not allowed to be used. A Positive list is a list that contains only the objects that have a privileged status.

13.2. StatusListOperation

The StatusListOperation describes what action the receiver should take after receiving a status list. It can either be Update, Delete or New.

```
StatusListOperation::=
     ENUMERATED {
          Update (0),
          Delete (1),
          New (2)
     }
```

Update means that the following list of objects, should be used to update the current list. This can either be adding new objects to the current list or updating existing entries in the list.

Delete means that the referenced objects should be deleted from the current list, and New means that the supplied list should replace the current entries in the list.

13.3. StatusListObject

The Status List object contains the characteristics of the objects that are to be put on or deleted from the status list. The `targetStart` and `targetEnd` can be referenced according to EntityID, prEN1545-1, ISO 9594, prENV12314-2 or ISO 7816-5

> NOTE: One object can also reference a range of card numbers. This is done by specifying a starting number and an ending number (inclusive).

The `startDate` and `endDate` are two optional elements that can be used to set up a starting date and an ending date in which the status listing is active.

The `properties` is also an optional element that can be used to set up for example a limit on the amount that can be purchased (purchase limit).

The `actions` is also an optional element that can be used to define what measures are to be taken when the defined conditions are fulfilled.

```
StatusListObject ::=
     SEQUENCE  {
          targetStart
               TargetReference,
          targetEnd
               TargetReference          OPTIONAL,
          startDate
               Date                     OPTIONAL,
          endDate
               Date                     OPTIONAL,
          properties
               Properties               OPTIONAL,
          actions
               Actions                  OPTIONAL
     }
TargetReference ::=
     CHOICE  {
          simpleID        [0]
               EntityID,
          cenTC224WG11ID [1]
               CENTC224WG11-ID     -- import from prEN1545-1,
          x501ID          [2]
               X501-ID,            -- import from ISO 9594
          cenTC278WG12ID [3]
               CENTC278WG12-ID     -- import from prENV12314-2
          iso7816-5-ID    [4]
               X501-ID,            -- import from ISO7816-5
     }
Properties ::=
     CHOICE  {
          purchaseLimit   [0]
               Amount,
          watermark       [1]
               Watermark
     }
```

```
Amount ::=
     SEQUENCE {
          value
               INTEGER,
          currency
               PrintableString
     }
Watermark ::=
     SEQUENCE {
          watermarkType
               INTEGER,
          watermarkContent
               ANY
                    DEFINED BY watermarkType
     }
Actions ::=
     CHOICE  {
          warning
               WarningLevel,
          disable
               DisableLevel,
          enable
               EnableLevel,
          record
               RecordLevel
     }
WarningLevel ::=
     INTEGER
DisableLevel ::=
     INTEGER
EnableLevel ::=
     INTEGER
RecordLevel ::=
     INTEGER
```

14. Payment

The information object Payment describes a payment within a transaction. It is a choice between different payment types.

```
Payment ::=
    SEQUENCE  {
        paymentType
            OBJECT IDENTIFIER,
        paymentContent
            ANY
                DEFINED BY paymentType
    }
```

paymentType is an object identifier that uniquely identifies the subsequent object. This could be an identifier that is defined in this Prestandard (see Annex A) or any other unique identifier. This means that the payment object can both be defined in this Prestandard as well as be defined elsewhere.

The paymentContent is the information object associated with paymentType. Two paymentContent objects are defined by this Standard.

14.1. SimpleAccountPayment

The SimpleAccountPayment object is used when a minimum of data is required. This could be for example a payment transaction where all other type of data is implicit or held at a central account.

```
SimpleAccountPayment ::=
    IMPLICIT SEQUENCE  {
        accountNumber
            AccountNumber,
        transactionAmount
            Amount,
        dateLocalTransaction
            NumericString
    }
```

The accountNumber is a number identifying the transaction. Its interpretation depends on the payment means used, i.e. it could be a prepaid card number or a bank account number.

```
AccountNumber ::=
    CHOICE  {
        simpleID          [0]
            EntityID,
        cenTC224WG11ID [1]
            CENTC224WG11-ID     -- import from prEN1545-1,
        x501ID            [2]
            X501-ID,            -- import from ISO9594
        cenTC278WG12ID [3]
            CENTC278WG12-ID     -- import from prENV12314-2
    }
```

14.2. BankCardPayment

The `BankCardPayment` data structure accommodates post-payment, pre-payment, on-board or central account and a variety of payment means e.g. credit card, debit card, etc. This data structure is used as a placeholder for payment objects based on ISO 8583.

14.3. wg9Payment

This data structure is used as a placeholder for payment objects defined in CEN/TC224.

14.4. wg10Payment

This data structure is used as a placeholder for payment objects defined in CEN/TC224.

14.5. wg11Payment

This data structure is used as a placeholder for payment objects defined in prEN1545.

15. PT03

The information object PT03 can be used to send data elements defined by CEN/TC278/WG1/PT03.

NOTE: Since these elements are not yet available, no explicit references to these elements can be made in this Prestandard

```
PT03 ::=
     SET OF
          ANY
```

Annex A (Normative): Object Identifiers

This annex contains a summary of the object identifiers defined in this Prestandard.

```
errorCode ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.errorCode
     }

reportSent ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.reportSent
     }

statusList ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.statusList
     }

payment ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.payment
     }

wg11 ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.prEN1545
     }

pt03 ::=
     OBJECT IDENTIFIER {
          CEN.TC278.AFC.prENV-ISO-14906
     }
```

Annex B (Informative): Conceptual Model

The basis for the interface specification defined in this Prestandard is a conceptual model which defines the entities present in a generic payment system along with the relations existing between these entities.

This model was developed to meet/satisfy/fulfil the following requirements/conditions/goals:

a) comply with the scope of the Prestandard;
b) remain valid in a great variety of situations, ranging from simple local transport systems involving only one service and one organisation to the most complex systems operating on a higher (e.g. pan-European) level and supporting many types of (chained) services, many service providers, many collection agents, many issuers and many clearing operators;
c) support different organisational models (subsidiarity principle);
d) retain compatibility with the models used in the financial world.

The flexible yet powerful conceptual model presented in figure B1 achieves these goals through the use of generic abstract entities instead of referencing real-world organisations. The transport world is characterised by a great diversity of business and organisational arrangements — in both existing and future systems — and only by relying on generic abstract entities can different types of organisational arrangements be expressed using the same generic model.

The entities identified and defined in the conceptual model constitute generic abstract entities and not organisations.

The main consequences of this approach are:

a) the conceptual model does not imply or mandate/require that there should always be a separate organisation for each abstract entity in every real-world system. Depending on the particular business arrangements and resulting organisational models, the generic abstract entities Clearing Operator and Collection Agent may or may not have direct counterparts in the real-world;
b) the conceptual model does not imply any particular scheme as to where and how long financial value (i.e. float associated with prepaid methods) is being held.
c) As an entity rather than a functional model, the conceptual model is the only model able to fully support the transport world while being highly compatible with the models used in the financial world (by the banking industry and the Intersector Electronic Purse).

The conceptual model presented in figure B1 defines the following generic abstract entities which compose the payment system and their relationships:

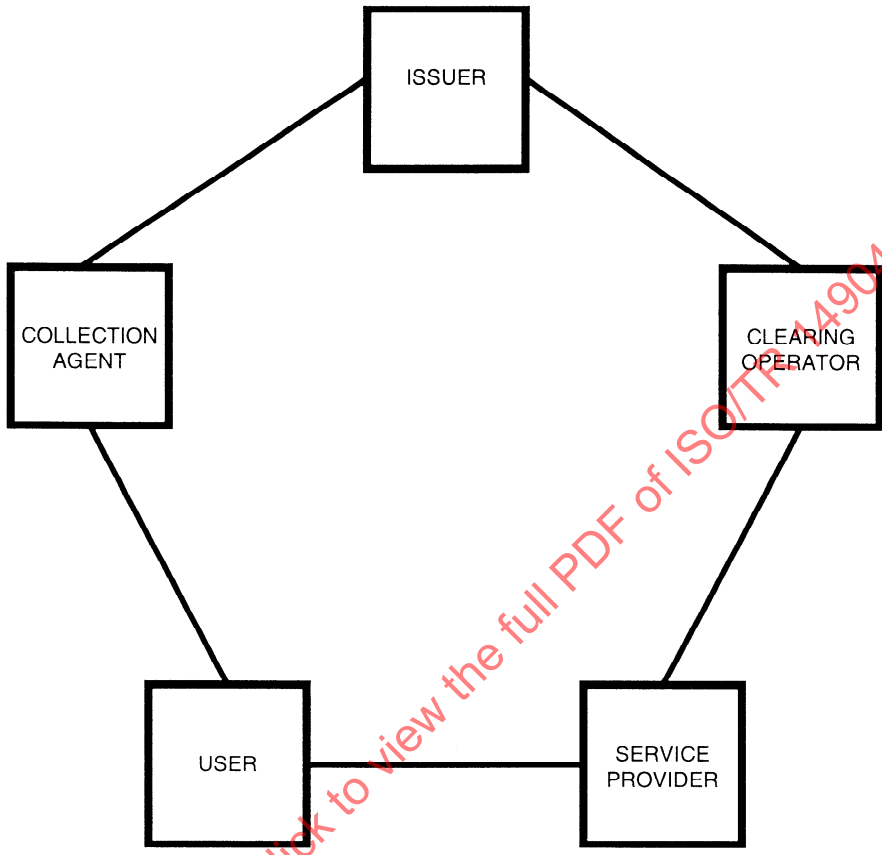

Figure B1 : Conceptual Model

This conceptual model combines the payment system models of the financial world and the payment systems models of the transport world. It supports both pre-payment and post-payment modes.

NOTE: The entities described in the model constitute generic abstract entities which can be mapped or combined into real-world organisations (see Annex C).

Figure B2 shows the basic interfaces identified within a payment system and between payment systems.

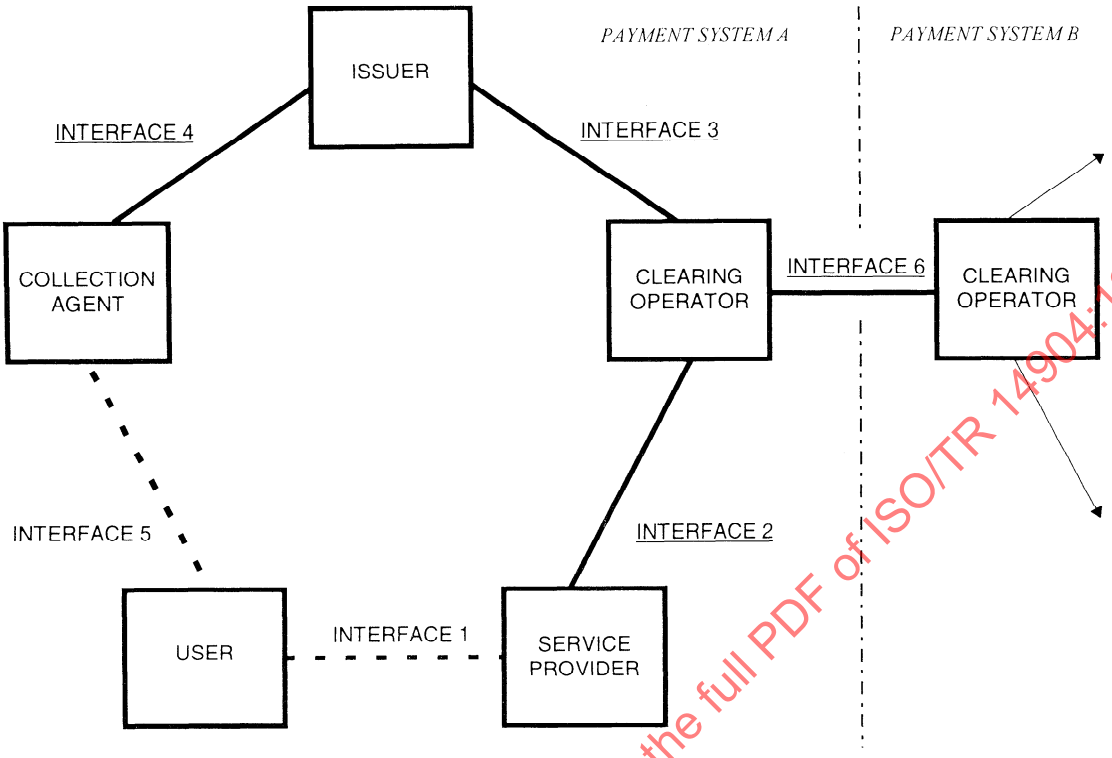

Figure B2 : Basic interfaces in a Payment System

NOTE 1: The interfaces covered by this Prestandard are those in continuous bold and underlined.

NOTE 2: While this model clearly identifies a basic Clearing Operator to Clearing Operator interface (interface 6), the interface specification support equally well information transfers for clearing or other purposes between any two entities within a payment system or between two payment systems.

Annex C (Informative): Relation between Conceptual Model and Organisational Models

Examples of mapping of generic abstract entities into real-world organisations

In the present political and economical context in Europe, it does not seem likely or even possible that a single organisational structure will satisfy the different contractual and economical requirements of every transport related AFC system. The subsidiarity principle — which states that responsibilities should be left at the level where they are managed— requires that the model selected should not impose unacceptable limits on the kind of contractual and organisational arrangements possible between operators within the payment system or between payment systems.

In order to remain valid in a wide range of situations the conceptual model describes a generic payment system using the complete set of generic abstract entities whereas real-world systems are built and operated by real — not virtual — organisations. In addition, not all payment systems require a Clearing Operator or a Collection Agent.

The correspondence between the conceptual model and a particular organisational model — which may be already used to describe an existing system — is achieved by mapping the generic abstract entities into the real-world organisations present in the system. This approach permits to support equally well a wide range of organisational models by clearly disconnecting the technical issues related to the interfaces from the business and financial arrangements.

This mapping means that the functions associated with the generic abstract entities of the conceptual model can be combined in many different ways to reflect the contractual and technical arrangements between the organisations involved in the payment system. This in effect means that the functions of one or several generic abstract entities can be combined and assumed by one organisation, e.g. an organisation which combines the functions associated with the Issuer and the Clearing Operator.

Those possible combinations range from the situation where there is a separate organisation for each generic abstract entity e.g. one company issues the physical device, another issues the payment means, another supplies the service, etc., to the situation where all generic abstract entities, except the user, are combined and their functions assumed by one organisation.