TECHNICAL SPECIFICATION

ISO/TS 14904

First edition
2002-12-15

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

Télématique de la circulation et du transport routier — Perception du télépéage — Spécification des interfaces pour la compensation des recettes entre opérateurs

Reference number
ISO/TS 14904:2002(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization 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 and 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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of normative document:

— an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;

— an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TS 14904 was prepared by the European Committee for Standardization (CEN) in collaboration with Technical Committee ISO/TC 204, *Transport information and control systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

Throughout the text of this document, read "...this European pre-Standard..." to mean "...this Technical Specification...".

This first edition of ISO/TS 14904 cancels and replaces ISO/TR 14904:1997, which has been technically revised.

Contents

Foreword

The text of ENV ISO 14904:2002 has been prepared by Technical Committee CEN/TC 278 "Road Transport and Traffic Telematics", the secretariat of which is held by NEN, in collaboration with Technical Committee ISO/TC 204 "Transport Information and Control Systems".

This European Prestandard supersedes ENV ISO 14094:1997.

In this European Prestandard, the annexes A to F are informative.

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

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 European Prestandard / ISO Technical Standard 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 European Prestandard / ISO Technical Standard 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.

1 Scope

This European Prestandard specifies 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 Electronic Fee Collection (EFC) 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 European 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 European 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 European Prestandard.

Not described within this European 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.

2 Normative references

This European Prestandard incorporates by dated or undated reference, provisions from other publications. 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 apply to this European Prestandard only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies (including amendments).

ISO/IEC 7812 (all parts), *Identification cards — Identification of issuers*

ISO/IEC 7816-5, *Identification cards — Integrated circuit(s) cards with contacts — Part 5: Numbering system and registration procedure for application identifiers*

ISO 8583, *Financial transaction card originated messages — Interchange message specifications*

ISO/IEC 8825-1, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

ISO 9594 (all parts), *Information technology — Open Systems Interconnection — The Directory*

ISO 11770-1, *Information technology — Security techniques — Key management — Part 1: Framework*

ENV ISO 14816, *Road transport and traffic telematics — Automatic vehicle and equipment identification — Numbering and data structure*

ENV ISO 14906, *Road Transport and Traffic Telematics (RTTT) — Electronic Fee Collection (EFC) — Application interface definition for dedicated short range communications*

ENV 1545-1, *Identification card systems — Surface transport applications — Part 1: General data elements*

3 Terms and definitions

For the purpose of this European Prestandard, the following terms and definitions apply.

3.1

Apportionment

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

3.2

Chained Services

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

3.3

Clearing

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*

3.4

Clearing Operator

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

3.5

Collection Agent

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*

3.6

Contract

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

3.7

(Intersector) Electronic Purse

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*

3.8

Electronic Fee Collection

collection of a fee for a transport service where the fee is collected via the exchange of data, e.g. via an air-link communication, enabling the user to pay for the service with electronic values, e.g. an electronic purse or values stored in a central account

3.9

Enforcement Operator

entity responsible for prosecution on the basis of violation information provided by the Service Providers.

3.10
Integrated Payment Systems
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*)

3.11
Issuer
entity responsible for the payment system and responsible for issuing the *Payment Means* to the *User*

3.12
Operator
generic term for the entities Issuer, Clearing Operator, Collection Agent, Service Provider, Enforcement Operator or Trusted Third Party

3.13
Payment Means
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*

3.14
Payment Method
combination of a Payment Means, a Payment Mode and a Payment Scope

3.15
Payment Mode
parameter defining the time dimension in payment by the *User, e*.g. Pre-payment or Post-payment

3.16
Payment Scope
application extent of the *Payment Method*, e.g. national transport or inter-sector

3.17
Payment System
financial system that includes the complete process of *Issuing,* use of *Payment Means*, *Clearing* and *Settlement* of transactions

3.18
Service Provider
person, company, authority or abstract entity offering a service to the *User* for which the user has to pay a fee (the fee can in some cases be zero, e.g. emergency vehicles)

3.19
Settlement
transfer of funds from one *Operator* to another according to the *Clearing* rules

3.20
Trusted Third Party
entity who might be responsible for operation monitoring, system and security assessment (including security key management) as well as granting licences

3.21
User
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*

4 Basic interfaces for clearing between operators

This European Prestandard identifies the following basic interfaces required for clearing between operators within a payment system and between payment systems (see annex A Conceptual Model for further explanations):

Table 1 – Overview of operator interfaces

Operators interfaced	Interfaces covered by the standard	Interfaces NOT covered by the standard
Any Operator to any Operator (see definition of Operator in 3)	X	-
User - Service Provider	-	X
Collection Agent – User	-	X

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

5 Interface framework

5.1 Introduction

Clause 5 defines a common message structure to enable the exchange of data on any of the interfaces between operators.

The common message structure is summarised in 5.2 and described in more detail in annex C.

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

5.2 Summary of message structure

The message structure shall be transferred either explicitly defined in this standard or implicitly using services defined by other communication protocols.

EXAMPLE TCP/IP, XML/EDIFACT can be used to transfer messages.

Figure 1 shows graphically an example of the message structure for the Electronic fee Collection (EFC) related Protocol Data Unit (PDU). The objects shown in the diagram (the information forming the Message Body) can either be unsecured or secured globally or individually.

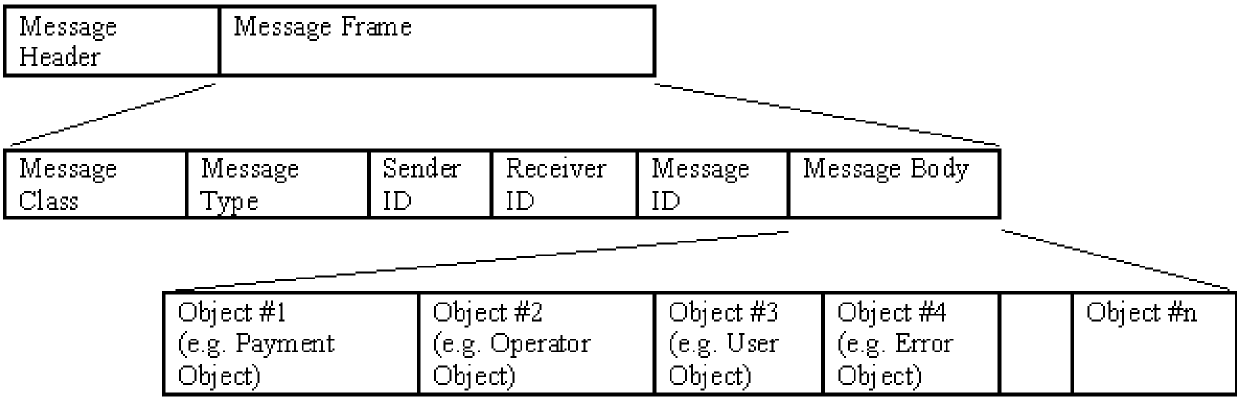

Message Header
Message Frame
Message Class
Message Type
Sender ID
Receiver ID
Message ID
Message Body
Object #1 (e.g. Payment Object)
Object #2 (e.g. Operator Object)
Object #3 (e.g. User Object)
Object #4 (e.g. Error Object)
Object #n

Figure 1 - Example of the message structure

5.3 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 is always the first element in the sequence, the receiving party is always able to identify the version of the protocol being used to send the data. This European Prestandard defines version 2 of the protocol.

NOTE ENV ISO 14904:1997 defines version 1 of the protocol.

5.4 Message frame

The message frame may be included in the message structure defined in 5.2. Annex C shows how the message frame can be formatted.

5.5 Security data

The main objective of Data Protection in EFC systems is to protect the interests of those relying on the EFC systems, from any harm or damage caused by lack of availability, confidentiality, integrity, non-repudiation and privacy of personal data.

Part of the information exchanged over the interfaces is covered by this European Prestandard, constituting an important asset for the respective parties involved. Whilst 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 should make sufficient provision to incorporate current and future security related items.

The security data at the message level and the secured data objects 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 can 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.

5.6 Security and Privacy

As EFC systems need to address both data security and privacy issues, defined in the following as a combined domain called Data Protection, their architecture needs also to provide the adequate support. In EFC system architectures, and for the purposes of this standard, privacy is taken as being related to the rights of individual users of the system in respect with the way their personal data is stored and handled within the EFC system and possibly across EFC systems, e.g. clearing between operators.

5.7 Data Protection Framework

The model shown in Figure 2 provides a general framework for interpreting the primary relationships between the main issues and elements involved in the planning design and operation of data protection schemes:

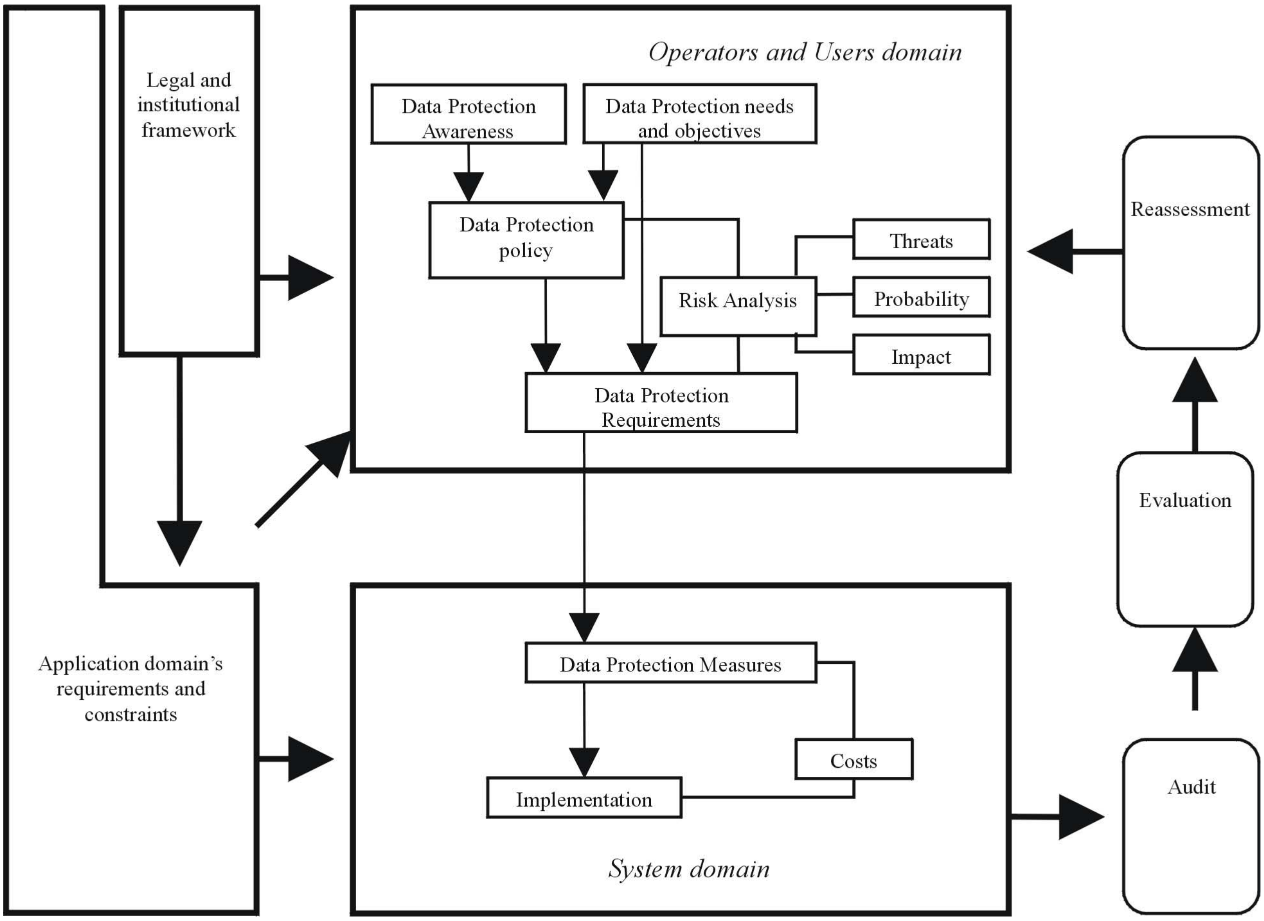

Legal and institutional framework
Operators and Users domain
Data Protection Awareness
Data Protection needs and objectives
Data Protection policy
Threats
Risk Analysis
Probability
Impact
Data Protection Requirements
Reassessment
Evaluation
Application domain's requirements and constraints
Data Protection Measures
Costs
Implementation
System domain
Audit

Figure 2 - Data protection framework

In the Operator and Users domain, a data protection policy is defined based on the overall needs and objectives of the operators and users of the EFC systems, the results of the risk analysis, and the awareness of the general issues involved in data protection (i.e. data protection principles).

The results of the risk analysis — which consists mainly in an evaluation of the possible threats to the EFC systems, their probability of occurrence and the possible impact — as well as the data protection policy and the overall needs and objectives, are used to define detailed and precise Data Protection Requirements.

These requirements are in turn used as the basis for the definition of the measures to be applied in the EFC systems to counter the threats or minimise their effect. In the associated process the constraints and additional requirements of the application domain, as well as the costs associated with the measures and their implementation — in accordance with the proportionality principle — are also taken into account when defining the countermeasures.

In addition, the legal and institutional framework, as well as the constraints and other requirements of the application domain need to be considered when establishing the data protection policy and data protection requirements for the system(s).

Finally, in accordance with the reassessment principle, the system in operation is subjected to auditing procedures, resulting in an evaluation and a reassessment of the threats, their probability and their impact.

5.8 Data Protection measures

Figure 3 gives an overview of a methodology for specifying the Data Protection:

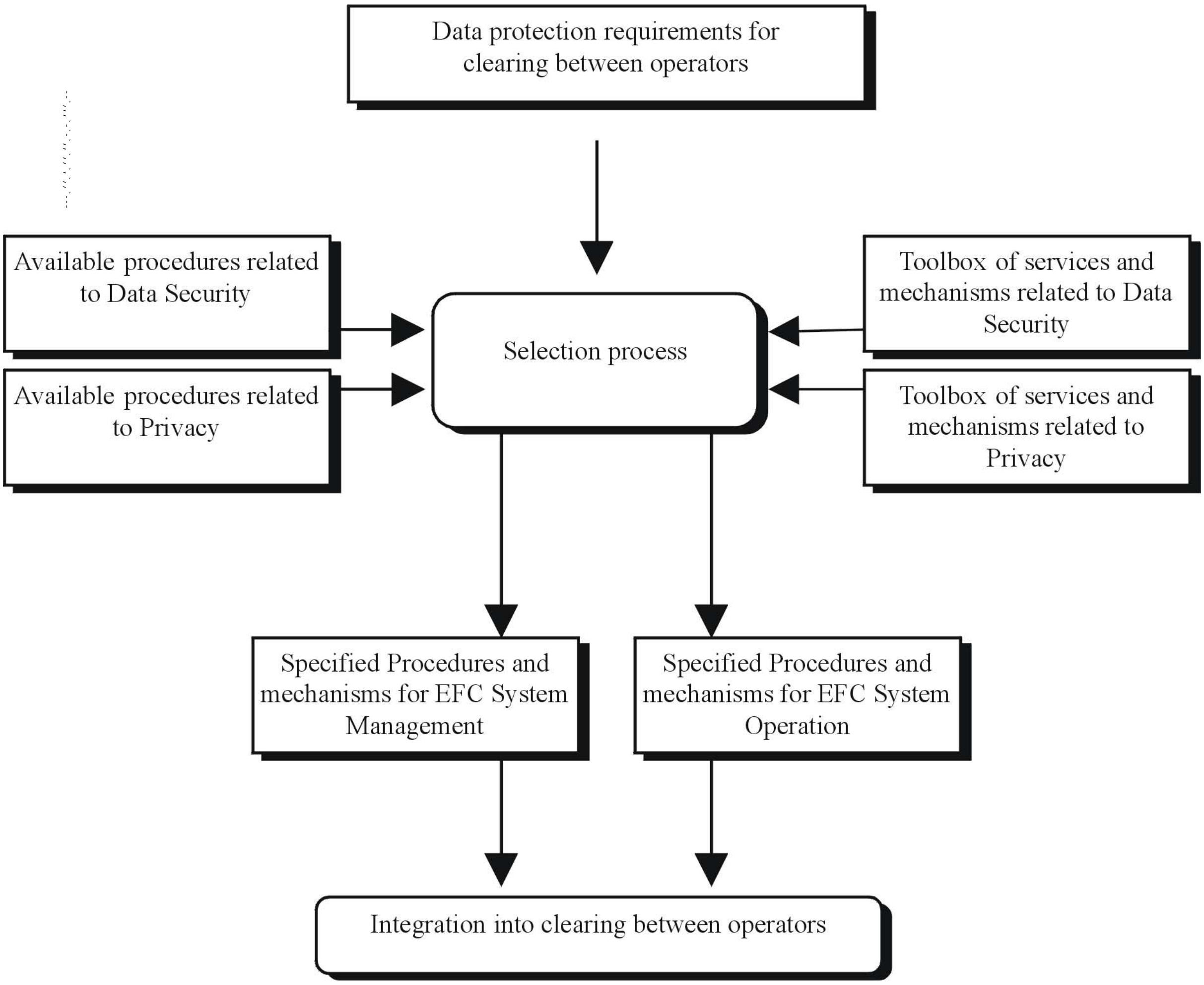

Data protection requirements for clearing between operators
Available procedures related to Data Security
Available procedures related to Privacy
Selection process
Toolbox of services and mechanisms related to Data Security
Toolbox of services and mechanisms related to Privacy
Specified Procedures and mechanisms for EFC System Management
Specified Procedures and mechanisms for EFC System Operation
Integration into clearing between operators

Figure 3 - Specification of data protection measures

5.9 Keys and keys management

This part provides a general introduction to the use and handling of keys and key management, which is an important part of clearing between operators. The description is according to ISO 11770-1.

5.9.1 Keys

Keys are a critical part in EFC systems when relying on cryptographic techniques. Keys have to be protected against disclosure, modification and deletion.

Keys are generally organised in hierarchies, where keys in one level of hierarchy may only be used to protect keys in the next level, while the lower keys are used for providing the security services. A security system normally consists of two types of keys:

1) keys that are used for encryption of data;

2) keys that are used for encryption of keys.

Generally the latter need more protection than the first. A so-called Secure Application Module (SAM) may provide secure storage of keying material.

A cryptographic key undergoes different phases in its life cycle, as shown in Figure 4:

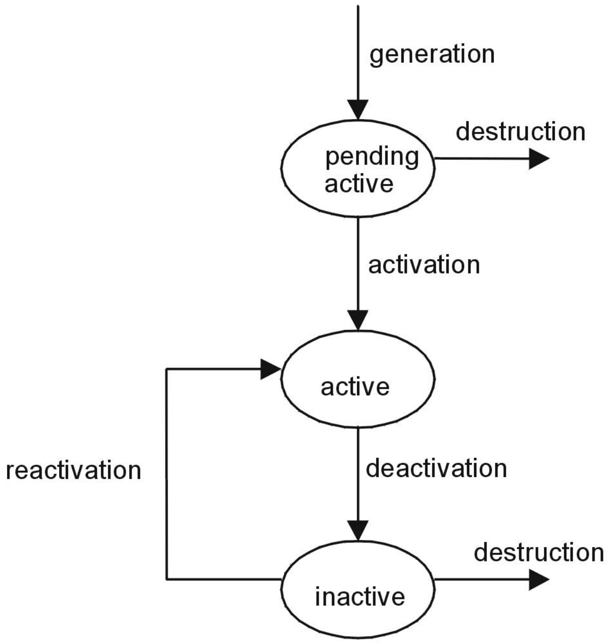


Figure 4 - The life cycle of a cryptographic key

5.9.2 Key management

The objective of key management is to provide secure administration of the key management services. The key management services are generation, registration, certification, de-registration, distribution, storage, archiving, recovery, deletion, derivation and destruction of cryptographic keying material.

As shown in Figure 5, several users may use the key management services. In EFC systems this includes first of all the Service Providers and the Trusted Third Party, but also the other entities are involved in mainly the distribution service.

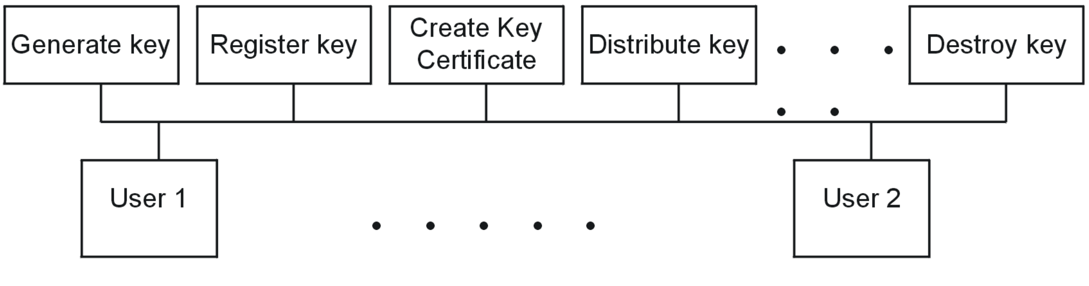


Figure 5 - Key management services

A key enters different states depending on the type of security and cryptographic system it consists of. This means that key management varies between symmetric and asymmetric techniques.

5.9.3 Key distribution

The distribution of keys can be done either within one security domain or between two security domains. Within one security domain the distribution may be done directly between the two entities that need to share keys, or it can be done through a Key Distribution Centre, which is a common security authority (e.g. TTP) that generates and distributes a common key between the two. This latter model may also be used when the entities belong to two different security domains if they trust the authority of one of the domains. One of the security authorities then generates and distributes the key to the respective authority of the other domain, see Figure 6.

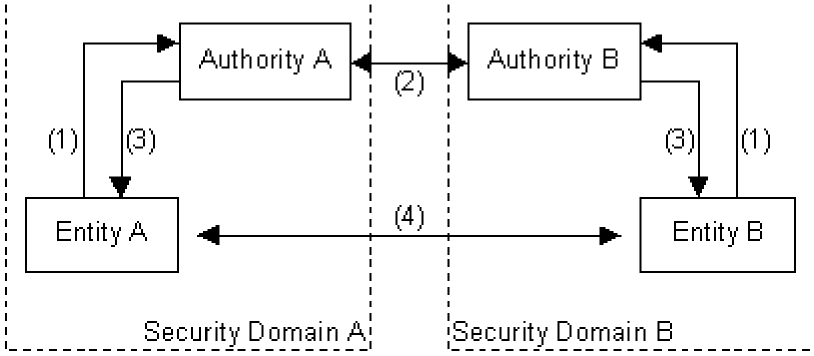


Figure 6 - Key distribution

6 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) as defined by ISO/IEC 8825-1 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 can be transmitted applying different encoding rules. 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 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.

7 Message

The `Message` type describes the complete data transferred between two operators. The data is contained within a sequence. See also 5.2 to 5.4 and annex C.

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

                                               version2(2) },
          data
               ProtocolDataUnit
     }
```

To identify different versions of the protocol, the message starts with an integer identifying the version. Since this integer is always the first element in the sequence, the receiving party is always able to identify the version of the protocol in use. ENV ISO 14904:1997 defines version 1 of the protocol. This European Prestandard defines version 2 of the protocol.

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

```
ProtocolDataUnit ::=
     CHOICE  {
          EFCrelated [0]
               EFCRelated-PDU,
          other
               EXTERNAL
     }
```

```
EFCRelated-PDU
     CHOICE {
          globallySecuredEFCData
               EXTERNAL

          locallySecuredEFCData
               LocallySecuredEFCData
     }
```

NOTE Locally Secured EFC 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 `locallySecuredEFCData` is the data structure defined in this European Prestandard.

The `EXTERNAL` data type allows for data types defined by other standards to be encapsulated in the `ProtocolDataUnit` used in this European 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 European Prestandard, the same communicating link can be used.

Since the data type is `EXTERNAL`, no interpretation of the data can be done using this European 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 European Prestandard does not define any identifiers for this data type `EXTERNAL`.

Annex A
(informative)

Conceptual Model

The basis for the interface specification defined in this European 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 fulfil the following goals:

a) comply with the scope of the European 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 A.1 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 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 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 A.1 defines the following generic abstract entities, which compose the payment system and their relationships:

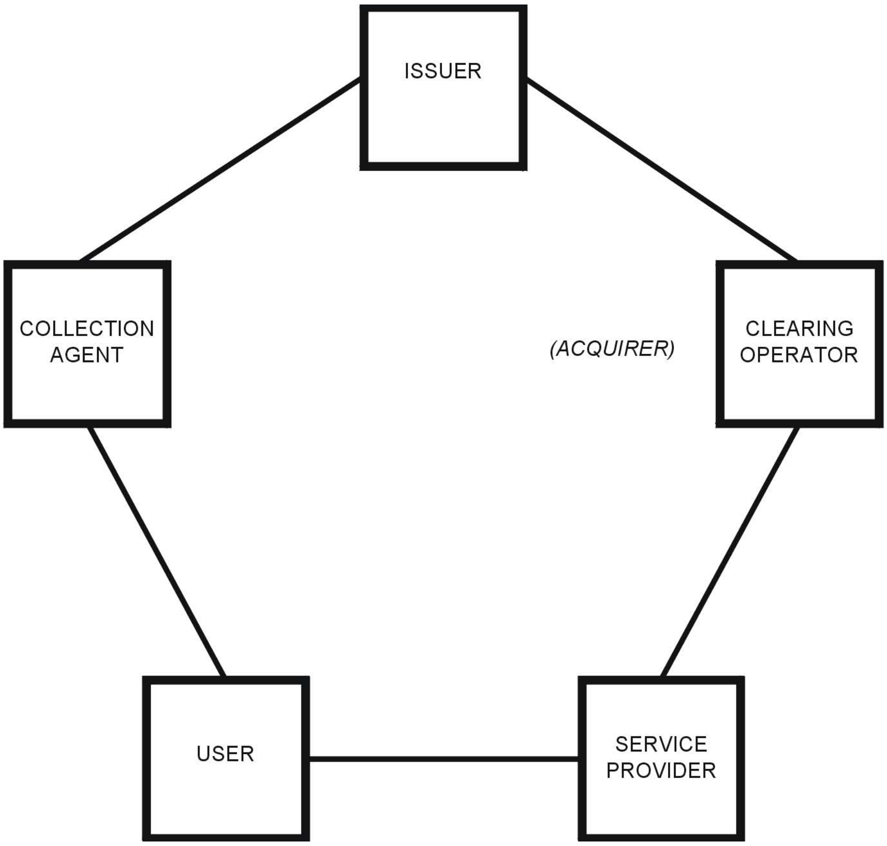


Figure A.1 - Conceptual model

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 1 The entities described in the model constitute generic abstract entities that can be mapped or combined into real-world organisations (see annex B).

NOTE 2 The interfaces primarily covered by this European Prestandard are those shown with continuous lines. However, in some cases exchange of information as shown with the dotted lines would also find this European Prestandard being applicable.

prENV ISO 17573 Road Transport and Traffic Telematics - Electronic Fee Collection - System architecture for vehicle related Transport services gives a more comprehensive description on how the generic payment system model can be applied.

Annex B
(informative)

Relation between Conceptual and Organisational Models

This annex provides examples of the 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 satisfies the different contractual and economical requirements of every transport related EFC 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.

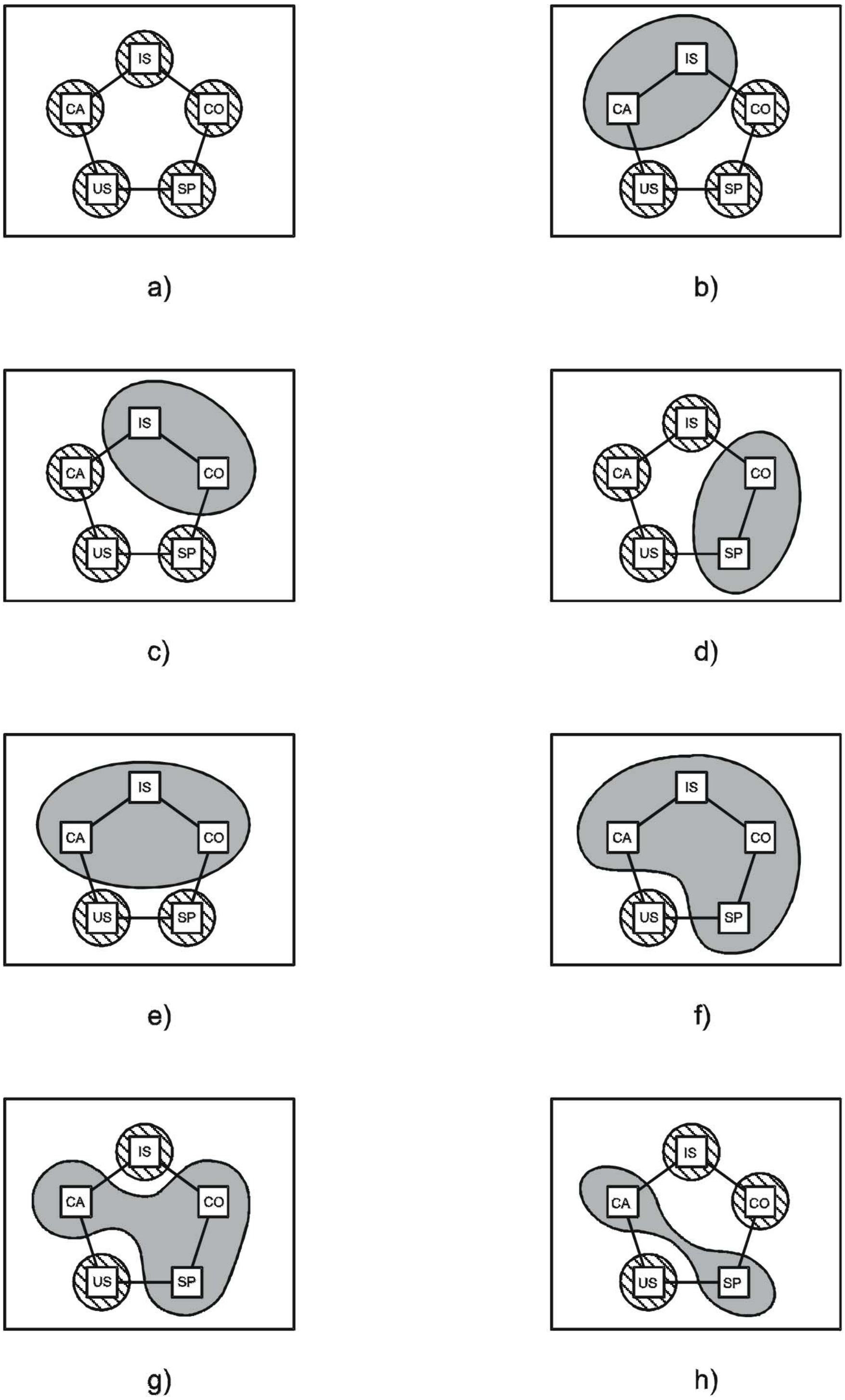
In order to remain valid in a wide range of situations the conceptual model describes a generic payment system using the complete set of generic abstract entities whereas real-world systems are built and operated by real — not virtual — organisations. In addition, not all payment systems require a Clearing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

The correspondence between the conceptual model and a particular organisational model used e.g. 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.

Figure B.1 shows some examples on different combinations. To simplify the figure the Trusted Third Party and the Enforcement Operator are not shown. The Enforcement operator is very often the Service Provider while the Trusted Third Party often is an external company or authority responsible for the security including the security key management. In that respect the Trusted Third Party is seen as an entity on an abstract level above the 5 entities shown in Figure B.1.



IS	Issuer	CO	Clearing Operator
CA	Collection Agent	US	User
SP	Service Provider		

Figure B.1 - Examples of mapping of generic abstract entities into real-world organisations

Figure B.1a) shows a situation where there is a separate organisation for each generic abstract entity which is not the User. Each separate organisation assumes the functions of its generic abstract entity counterpart. The Service Provider is responsible only for supplying the service and accepting the payment means.

Figure B.1b) shows a situation where one organisation combines and undertakes the functions associated with the Collection Agent and the Issuer. The Service Provider is responsible only for supplying the service and accepting the payment means.

Figure B.1c) shows a situation where one organisation combines and undertakes the functions associated with the Issuer and the Clearing Operator. In such case there is no need for a Clearing operator and only its functions remain. The Issuer processes its own transactions. The Service Provider is responsible only for supplying the service and accepting the payment means.

Figure B.1d) shows a situation where one organisation combines and undertakes the functions associated with the Clearing Operator and Service Provider. An example of this in the real world would be where the Service Provider operates its own clearing function to send data to the relevant payment means Issuer.

Figure B.1e) shows a situation where one organisation combines and undertakes the functions associated with the Issuer, Clearing Operator and Collection Agent. In such case there is no need for a Clearing operator and only its functions remain. This organisation has complete control over the payment means. The Service Provider is responsible only for supplying the service and accepting the payment means.

Figure B.1f) shows a situation where all functions of the payment system are combined and operated by one entity. This would be most common in a single service single Service Provider environment. This is a description of a closed payment system.

Figure B.1g) shows a situation where one organisation combines and undertakes the functions associated with the Collection Agent, Clearing Operator and Service Provider functions. In such case there is no need for a Clearing operator and only its functions remain. An example of this would be where a multi-service payment means i.e. a city card is accepted for payment by a road toll operator. The road toll company accepts the payment means and also has the facility to re-load the User’s medium.

Figure B.1h) shows a situation where one organisation combines and undertakes the functions associated with the Collection Agent and the Service Provider. An example of this can be a public transport company that operates a transport service and has a network of ticket agents.

Figure B.2 gives an illustration of how individual payment systems can inter-communicate to form a higher (e.g. pan-European) level integrated payment system. Trusted Third Parties and Enforcement operators are not shown.

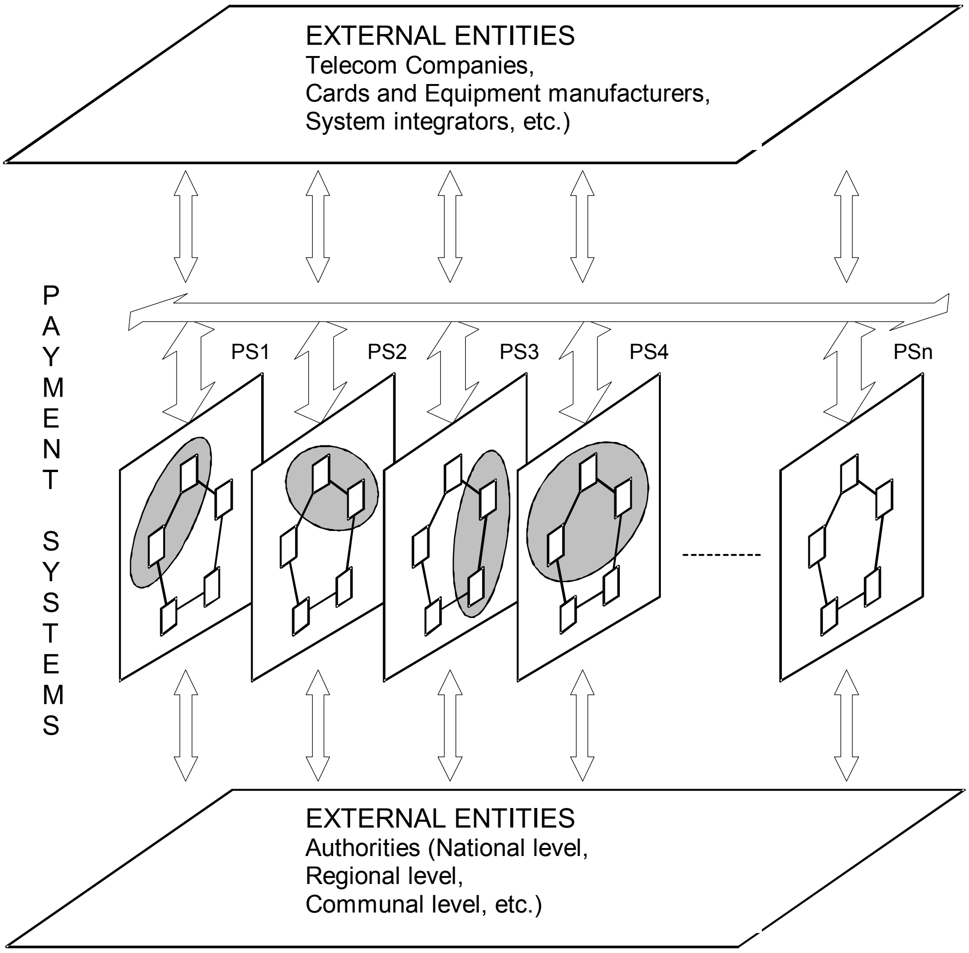


Figure B.2 - Interoperable payment system model

Figure B.2 shows a situation in which individual payment systems, described using the conceptual model, inter-operate. The diagram shows that the payment systems may be in contact with various external entities. These external entities may provide the communication network or be a local authority which provides the licence to operate. The model does not imply that each payment system can only communicate with other payment systems through Issuers or Clearing Operators but shows the concept of individual payment systems communicating on a higher (e.g. pan-European) level. The flexible nature of this European Prestandard allows for direct communications between operators in different payment systems.

Annex C
(informative)

Message frame format

C.1 Introduction

The message Frame 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:

- EFC related PDU (as described in this European Prestandard;
- any other PDU that is defined externally.

NOTE External PDUs can include: ISO 8583 PDU, CEN/TC224 PDU, EDIFACT PDU etc.

The Message Frame for the EFC 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 C.2 to C.7.

C.2 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 European Prestandard are:

C.2.1
Request
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

C.2.2
Request Response
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

C.2.3
Advice

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

C.2.4
Advice Response

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

C.2.5
Notification

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

C.2.6
Notification Acknowledgement

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

C.3 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 European Prestandard are:

C.3.1
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.

C.3.2
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.

C.3.3
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 effects 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.

C.3.4
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 apply 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 can also be used to inform the Clearing Operator of the apportionment rules that apply when operating the clearing process. These may include the discounts to be applied in the case of chained services.

C.3.5
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 then includes the totals.

C.3.6
Authorisation
this message type is used to obtain on-line authorisation from the Issuer for a particular transaction

C.3.7
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.

C.3.8
Report Sent
this message type is used by the sender to inform the receiver that a report has been sent to an external authority

C.3.9
Key Management
this message type is used to send keys or other security related information

EXAMPLE It can 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.

C.3.10
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) informs the receiver of the message on what to do with the status list

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

C.3.11
Equipment Status
this message type is used by the sender to send and/or request information on equipment status

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

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

C.3.12
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 can be used to send the data regarding the event to be enforced. This could include a video picture of the event.

C.3.13
Payment Method Acceptance

this message type is used to send 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.

C.3.14
Undefined Message Type

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

C.4 Sender ID

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 (any operator). 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.

C.5 Receiver ID

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 (any operator). 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.

C.6 Message ID

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 should contain the same message identifier.

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

C.7 Message Body

The Message Body is a sequence of data objects.

Annex D
(informative)

Protocol Data Unit

D.1 General

The `EFCrelated-PDU` introduced in clause 7, has a data structure `LocallySecuredEFCData` with the following definition:

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

D.2 MessageClass

All messages belong to one of six different message classes.

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

D.3 MessageType

D.3.1 General

The message type is the basic selection between different types of messages.

```
MessageType ::=
     ENUMERATED  {
          ServicesList (1),
          FareProductsList (2),
          CustomerDetails (3),
          ApportionmentRules (4),
          ReconciliationTotals (5),
          Authorisation (6),
          Payment (7),
          ReportSent (8),
          KeyManagement (9),
          StatusList (10),
          EquipmentStatus (11),
          EventException (12),
          PaymentMethodAcceptance (13),
          PrivatUse (14),
          Undefined Message Type (15),
          Undefined Message Type (16),
          Undefined Message Type (17),
          Undefined Message Type (18),
          Undefined Message Type (19)
}
```

D.3.2 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 uniquely identifies the subsequent object. This could be any unique identifier defined in other standards or agreed upon between operators.

The `paymentContent` is the information object associated with `paymentType.` Two paymentContent objects are defined by this European Prestandard / ISO Technical Standard.

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 ENV1545-1,
          x501ID          [2]
               X501-ID,               -- import from ISO9594
          isoTC204WG4ID    [3]
               ISOTC204WG4-ID         -- import from ENVISO14816-2
     }
```

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.

wg9Payment
This data structure is used as a placeholder for payment objects defined in CEN/TC224;

wg10Payment
This data structure is used as a placeholder for payment objects defined in CEN/TC224;

wg11Payment
This data structure is used as a placeholder for payment objects defined in ENV 1545.

D.3.3 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.

D.3.4 StatusList

The StatusList object is used to transmit status information. The status list can 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
     }
```

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.

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.

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, ENV 1545-1, ISO 9594, ENV ISO 14816-2 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 ENV1545-1,
          x501ID          [2]
               X501-ID,             -- import from ISO9594
          isoTC204WG4ID   [3]
               ISOTC204WG4-ID       -- import from ENVISO14816-2
          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
```

D.4 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]
               OBJECT IDENTIFIER ,
          cenTC224WG11ID [1]
               OBJECT IDENTIFIER    -- import from ENV1545-1,
          x501ID [2]
               OBJECT IDENTIFIER ,  -- import from ISO9594
          isoTC204WG4ID [3]
               OBJECT IDENTIFIER    -- import from ENVISO14816-2
     }
```

EntityID
The `EntityID` is an enumerated type identifying one of the entities defined in 3. 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)
          Trusted Third Party (6)
          Enforcement Operator (7)
     }
```

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

CENTC224WG11-ID
The CENTC224WG11-ID is used when an identity can be referenced through the use of a numeric string based on ENV1545-1.

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

ISO9594
The X501-ID is used when an identity can be referenced through the numbering schemes based on ISO 9594.

ISOTC204WG4-ID
The ISOTC204WG4-ID is used when an identity can be referenced through the numbering schemes based on ENV ISO 14816-2.

D.5 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
```

D.6 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 any unique identifier defined in other standards or agreed upon between operators.

The `messageObjectContent` is the information object associated with `messageObjectType`.

Annex E
(informative)

Payment Objects based on data elements defined in ISO 8583

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.

The BankCardPayment data structure is based on financial payment data elements defined in ISO 8583. The tag numbers of the object elements and their value types and lengths are defined by the bit map positions, value types and lengths of the corresponding ISO 8583 data elements.

```
BankCardPayment ::=
     IMPLICIT SET  {
          primaryAccountNumber                      [2]
               IMPLICIT NumericString (SIZE(1..19),
          processingCode                            [3]
               IMPLICIT NumericString (SIZE (6)),
          transactionAmount                         [4]
               IMPLICIT NumericString (SIZE (12)),
          cardholderBillingAmount                   [6]
               IMPLICIT NumericString (SIZE (12))          OPTIONAL,
          cardholderBillingConversionRate           [10]
               IMPLICIT NumericString (SIZE (8))           OPTIONAL,
          systemTraceAuditNumber                    [11]
               IMPLICIT NumericString (SIZE (6))           OPTIONAL,
          datetimeLocalTransaction                  [12]
               IMPLICIT NumericString (SIZE (12)),
          dateEffective                             [13]
               IMPLICIT NumericString (SIZE (4))           OPTIONAL,
          dateExpiration                            [14]
               IMPLICIT NumericString (SIZE (4))           OPTIONAL,
          dateConversion                            [16]
               IMPLICIT NumericString (SIZE (4))           OPTIONAL,
          pointOfServiceDataCode                    [22]
               IMPLICIT PrintableString (SIZE (12)),
          cardSequenceNumber                        [23]
               IMPLICIT NumericString (SIZE (3)),
          functionCode                              [24]
               IMPLICIT NumericString (SIZE (3)),
          cardAcceptorBusinessCode                  [26]
               IMPLICIT NumericString (SIZE (4)),
          acquiringInstitutionIdentificationCode [32]
               IMPLICIT NumericString (SIZE (1..11))       OPTIONAL,
          primaryAccountNumberExtended              [34]
               IMPLICIT PrintableString (SIZE(1..28)),
          serviceCode                               [40]
               IMPLICIT NumericString (SIZE (3)),
          cardAcceptorTerminalIdentification        [41]
               IMPLICIT PrintableString (SIZE (8))   OPTIONAL,
          cardAcceptorIdentificationCode            [42]
               IMPLICIT PrintableString (SIZE (15))  OPTIONAL,
          cardAcceptorName                          [43]
               IMPLICIT PrintableString (SIZE (1..99))     OPTIONAL,
          currencyCodeTransaction                   [49]
               IMPLICIT PrintableString (SIZE (3))   OPTIONAL
     }
```

Bibliography

ISO/IEC 7498-1:1994, *Information technology — Open Systems Interconnection — Basic Reference model: The Basic Model*

ISO/IEC 8824-1:1995, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 8825-2:1996, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*

ISO 9735, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (amended and reprinted)*

ISO 14813-6:2000, *Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 6: Data presentation in ASN.1*

ICS 03.220.20; 35.240.60

Price based on 31 pages