ETSI TECHNICAL REPORT

ETR 173

March 1995

Source: ETSI TC-TE

Reference: DTR/TE-10002

ICS: 33.020, 33.040.40

Key words: Multimedia, functional model

Terminal Equipment (TE); Functional model for multimedia applications

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Contents

Blank page

Foreword

This ETSI Technical Report (ETR) was produced by the Terminal Equipment (TE) Technical Committee of the European Telecommunications Standards Institute (ETSI).

ETRs are informative documents resulting from ETSI studies which are not appropriate for European Telecommunication Standard (ETS) or Interim European Telecommunication Standard (I-ETS) status. An ETR may be used to publish material which is either of an informative nature, relating to the use or the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

Blank page

1 Scope

This ETR provides a functional model to identify:

- functional units;
- interfaces between them; and
- interrelations between them

for any kind of multimedia application or service within a communication environment. The physical implementation of the functional units, their location in a real configuration and if a certain functional unit is necessary at all, depends on the particular multimedia application or service under consideration.

The functional model is intended to be used as a guide for further development within the ETSI multimedia project as follows:

a) the functional model should be complemented by a subsequent lower-level specification, i.e. all borderlines between functions will be retained and refined;

b) only when the functional model is found not to be feasible, should it be changed and published in a new version.

Thus, this functional model is hypothetical with respect to its technical feasibility.

Specific aspects such as network related aspects, encoding aspects, network management or any other aspects directly related to a specific implementation or identification of business partners involved in a general multimedia system are outside the scope of this ETR.

Work related to multimedia is going on in many organizations. This ETR does not pre-judge which organisation might be responsible for addressing the standardization of any particular aspect.

2 References

This ETR incorporates by dated or undated reference, provisions from other publications. These 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 ETR only when incorporated in it by amendment or revision. For undated references the latest edition of the application referred to applies.

[1] ETR 181: "Multimedia portfolio; A compilation of multimedia applications and services provided by ETSI members".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETR, the following definitions apply:

Application Platform: Collection of hardware and software components that provide system services to the application software. It is generic in nature and not visible for application software. It is serving application software only by means of messages communicated through the Application Programmable Interface (API).

Application Programmable Interface (API): Interface between the application software and the application platform across which all services are provided by the application platform. It is constituted by all communication that can be exchanged.

Application Software: Software that provides an application service to the user. It is specific to an application in the multimedia and/or hypermedia domain and is composed of programs and data.

hypermedia: The ability to access monomedia and multimedia information by interaction with explicit links.

local application: A piece of software which is part of multimedia and/or hypermedia applications and is residing physically in the considered equipment.

model: A generally accepted representation of a particular domain that allows people who are interested in that domain to build a fundamental understanding within the scope of the domain.

multimedia and hypermedia application: An application which involves the presentation of multimedia information to the user and interactive navigation across this information by the user.

multimedia application: An application which involves the presentation of multimedia information to the user.

multimedia object: A composite object composed of various different types of related temporal and logical content intended for presentation to the user.

multimedia: The property of simultaneously handling multimedia objects.

NOTE: The term multimedia is an adjective to be attached to a noun which provides the context: e.g. multimedia service or application, multimedia terminal, multimedia network, multimedia presentation.

Multipoint Control Unit (MCU): A piece of equipment located in a node of the network which connects terminals and, according to certain criteria, processes audiovisual signals and distributes them to the connected terminals.

object: A finite, independent self-defining piece of information container for content and structure that can be manipulated as a whole by applications and interchanged as one unit.

scripts: In the context of this ETR, scripts are programs written in a scripting language whereby scripting languages are subsets of programming languages. Scripts are interchanged as specific data content of objects used to develop multimedia applications. To execute scripts a script interpreter is needed.

Service Support Function: A functional unit which handles service specific control parameters. The functionality offered depends on the particular service.

3.2 Abbreviations

For the purposes, of this ETR, the following abbreviations apply:

API	Application Programmable Interface
CPSII	Communication Protocol Stack Independent Interface
ICF	Interconnection Function
JPEG	Joint Photographic Experts Group
M&H	Multimedia & Hypermedia
M&HIRS	Multimedia & Hypermedia Information Retrieval Services
MCU	Multipoint Control Unit
MHEG	Multimedia and Hypermedia Information coding Experts Group
MPEG	Moving Picture Experts Group
OCAM	Object Content Access Module
SIR	Script Interchange Representation
VCR	Video Cassette Recorder
VEMMI	Videotex Enhanced Man-Machine Interface

4 General model

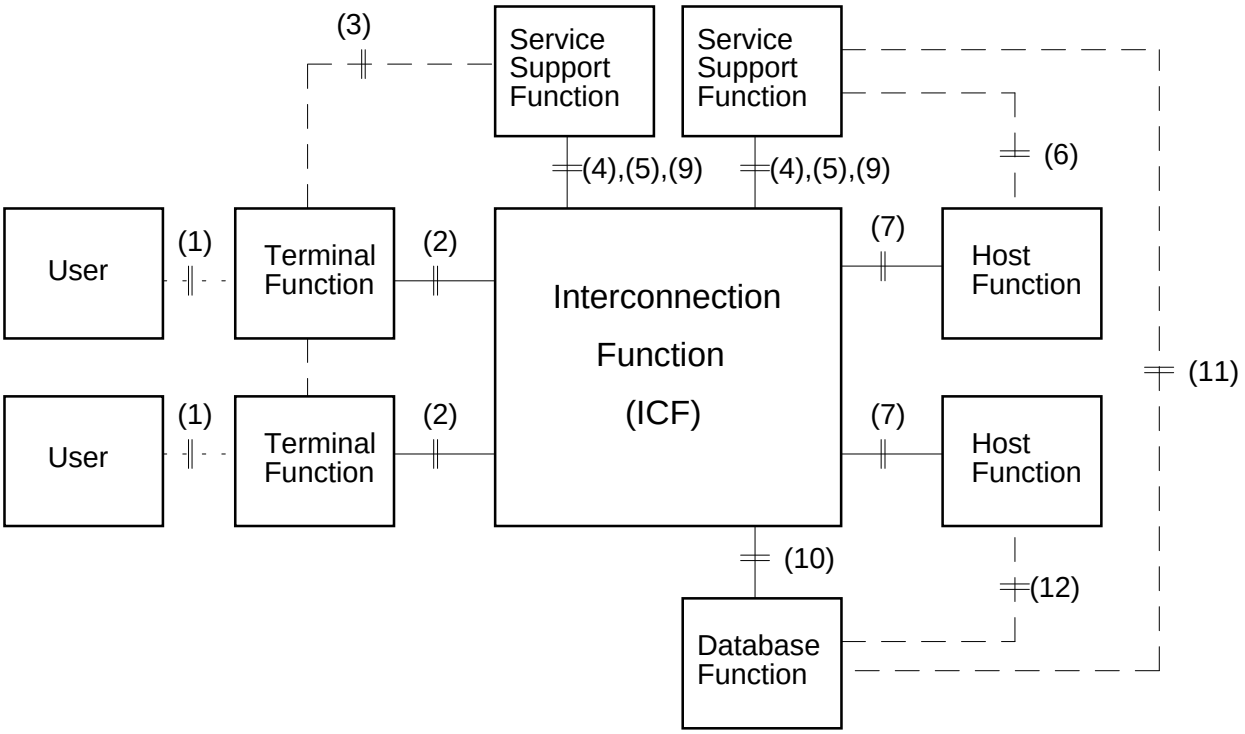
This clause defines a functional model describing a functional architecture common to the targeted multimedia applications. The model is applicable to all applications that interchange or process multimedia and/or multimedia/hypermedia objects.

This model is independent of the way of encoding multimedia and/or multimedia/hypermedia objects.

Figure 1 shows the reference model which is valid for:

- retrieval;
- conversational; and
- distribution

multimedia applications and services or also for combinations of the three categories as described in ETR 181 [1].



NOTE 1: The Interconnection Function (ICF) provides for a function access to/from other functions by means of one or more networks.

NOTE 2: Network management is not considered in the functional model.

Figure 1: Reference model

In the following, some cases of multimedia/hypermedia services are considered and it will be described which functional units (as mentioned in figure 1) are involved.

a) In the case of **multimedia/hypermedia retrieval services** the following functional units (and interfaces/protocols) are involved:

- User A ("Information Consumer" Man-Machine Interface (1));
- Terminal Function A (Network dependent protocols (2), end-to-end protocols (3));
- ICF (Network dependent protocols (2) in relation to the "Terminal Function A" and Network dependent protocols (4) in relation to the Service Support Function);
- Service Support Function (in this case "Access Function" with Network dependent protocols (4) and end-to-end protocols (3) in relation to the "Terminal Function A", and Network dependent protocols (5) and end-to-end protocols (6) in relation with the "Host Function");
- ICF (Network dependent protocols (5) in relation to the Service Support Function and Network dependent protocols (7) in relation to the Host Function);
- Host Function (Network dependent protocols (7) and end-to-end protocols (6));
- Database Function (this function can be merged with the Host Function. If not, then the end-to-end protocols (12) have to be used (i.e. functionality of end-to-end protocols (6) and (11) by using the ICF);
- User B ("Information Provider" Man-Machine Interface (1));
- Terminal Function B (Network dependent protocols (2), end-to-end protocols (3));
- Service Support Function (in this case "Access Function" with Network dependent protocols (4) and end-to-end protocols (3) in relation to the "Terminal Function B", and Network dependent protocols (9) and end-to-end protocols (11) in relation to the Database Function).

b) In the case of **conversational multimedia/hypermedia service for a point-to-point configuration** the following functional units (and interfaces/protocols) are involved:

- User A (Man-Machine Interface (1));
- Terminal Function A (Network dependent protocols (2), end-to-end protocols (8));
- ICF (Network dependent protocols (2) in relation to the Terminal Functions A and B);
- Terminal Function B (Network dependent protocols (2), end-to-end protocols (8));
- User B (Man-Machine Interface (1)).

c) In the case of **conversational multimedia/hypermedia service for a conferencing configuration** the following functional units (and interfaces/protocols) are involved:

- User A (Man-Machine Interface (1));
- Terminal Function A (Network dependent protocols (2), end-to-end protocols (3));
- ICF (Network dependent protocols (2) in relation to the "Terminal Function" and Network dependent protocols (4) in relation to the Service Support Unit);
- Service Support Function (in this case a MCU with Network dependent protocols (4) and end-to-end protocols (3) to the "Terminal Function" of all participating Terminals (e.g. A, B, etc. and/or to a cascaded MCU)).

d) In the case of a **multimedia distribution service** the following functional units (and interfaces/protocols) are involved:

- User A (in this case "Information providing user" Man-Machine Interface (1));
- Terminal Function A (Network dependent protocols (2), end-to-end protocols (3));
- ICF (Network dependent protocols (2) in relation to the "Terminal Function" and Network dependent protocols (4) in relation to the Service Support Unit);
- Service Support Function (Network dependent protocols (4) and end-to-end protocols (3) in relation to the "Terminal Function", and Network dependent protocols (9) and end-to-end protocols (11) in relation to the "Database");
- ICF (Network dependent protocols (9) in relation to the Service Support Function and Network dependent protocols (10) in relation to the Database);
- Database Function (Network dependent protocols (10) and end-to-end protocols (11));
- Terminal Function B (Network dependent protocols (2), end-to-end protocols (3));
- User B (in this case "Information Receiving User" Man-Machine Interface (1)).

As an example the possible location of necessary entities in a multimedia distribution service could be as follows.

EXAMPLE:	Information providing entity:	Terminal Function A
	Information selection entity:	Service Support Function
	Information distribution entity:	Database Function
	Information receiving entity:	Terminal Function B

As an example, for retrieval services, figure 1 can be refined taking into account different networks involved within the ICF (see figure 2).

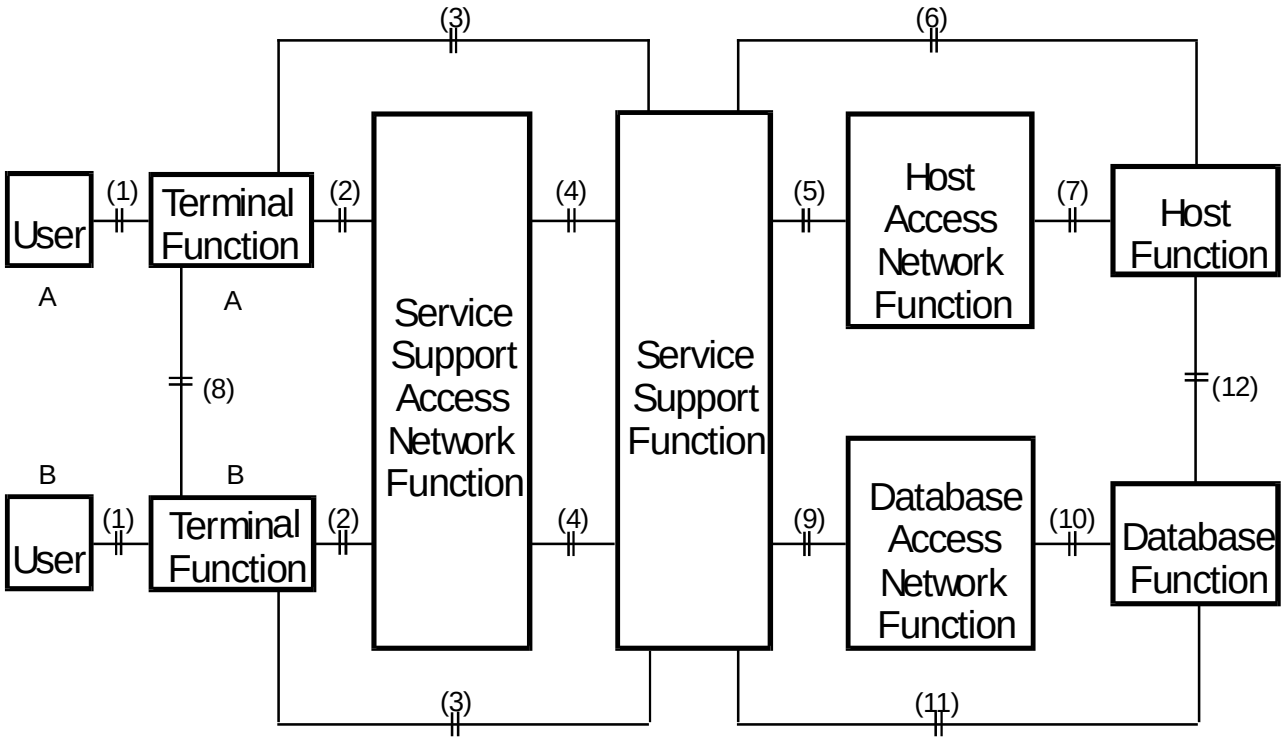


Figure 2: Reference model for retrieval services

Further examples of some of the possible configurations are given in annex B.

5 Functional units

Certain functional units contained within the reference model (as in figure 1) are described in the following subclauses, in order to give further levels of detail in these functional units.

5.1 Terminal function

The conceptual overview of the "Terminal Function" is given in figure 3:

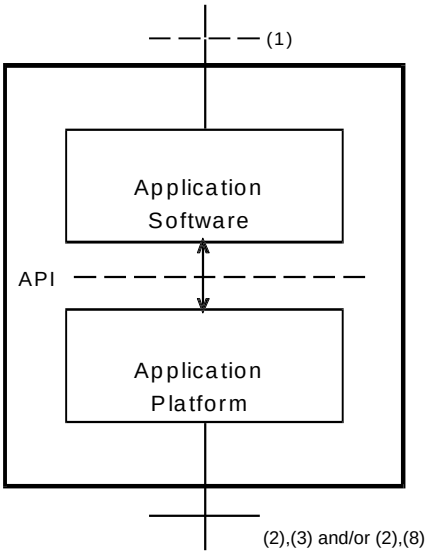


Figure 3: Conceptual overview (Terminal Function)

The Terminal Function comprises the Application Software, the Application Platform and the Application Programmable Interface (API) between them.

Depending on the multimedia service under consideration different categories of networks will be most suitable (narrow band networks, broadband networks).

Figure 4 shows the "Terminal Function" in more detail (further examples are given in annex A).

Figure B.1 in annex B uses an example to explain the role of the interface identifiers as the set (1), (2), (3), (2), (8) above.

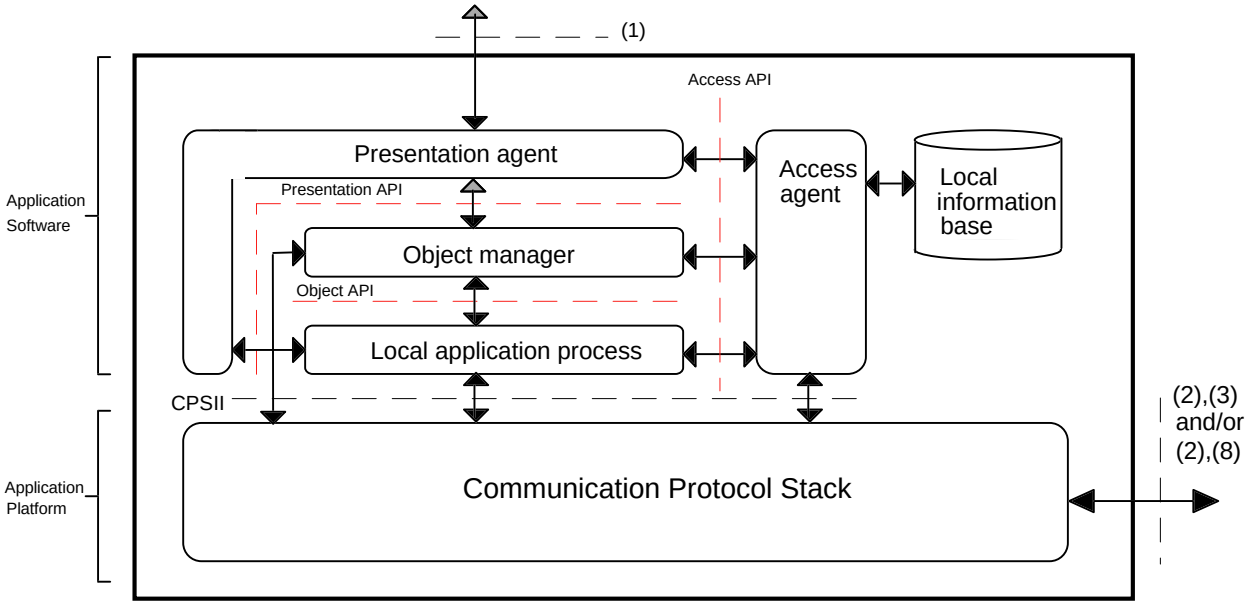


Figure 4: Terminal Function

Various instances for **Communication Protocol Stacks** are possible depending on the specific needs of multimedia services.

The Communication Protocol Stack Independent Interface (CPSII) provides a network independent access to the used network. The end-to-end protocols can also be different, depending on the specific multimedia services used, and therefore this CPSII is important for the possibility to build "Common blocks" above the Communication Protocol Stack.

The **presentation agent** manages the presentation of monomedia data, performs data format decoding and manages the interaction. It also acts as an interface to devices external to the multimedia functional model. However, their functions are assumed by multimedia service specifications, e.g. smartcard readers, Video Cassette Recorders (VCRs). The presentation agent is accessed by its clients through the "presentation API" which isolates the software on higher level from the specific features of the various hardware sub-platforms; the main components of the presentation agent are JPEG decoders, MPEG decoders, Videotex syntax decoders as well as decoders and encoders from input devices (keyboard, mouse, voice recognition, etc.). The contents to be presented following a request through the presentation API are normally specified by reference identification; the reference is input to the access agent.

The **access agent** manages a directory that enables it to locate the requested content. The content is either on the local object base because it is resident or has been previously downloaded, or it is located on a distant memory (distant multimedia database or cache memory in the network); in this last case the access agent automatically sends the appropriate request through the CPSII to get the object or the content. The access agent makes the localisation of the various objects (multimedia contents, MHEG objects, scripts, application specific data) totally transparent to its clients (the processes trying to access them).

The **local application** manages the logic of a given platform. The application itself will often be distributed between several platforms (terminals, hosts). The local application is a client to the access agent via the access API, of the object manager via the object API and of the presentation agent via the presentation API. The local application can be very simple (pass through to the presentation agent) or may include in some cases a **script processor** allowing execution of scripts, i.e. parts of the application which are interchanged during the course of an application. The script processor is a functional element in charge of executing scripts. It may also be the script itself (if interchanged in executable form), a script language interpreter or a Script Interchange Representation (SIR) interpreter.

The object manager interprets objects, manages links between them, triggers actions and orders objects presentation and access and provides temporal and spatial synchronisation between multimedia and hypermedia objects. It is controlled by the application process through the object API. Just as with the presentation server, the object manager uses the access agent each time it needs to process a referenced object.

The **local information base** may be used to store objects, contents, scripts, etc., permanently or temporarily on the device. No assumption is made on how those objects are stored and which physical storage device is used.

The **presentation API** allows the terminal object manager to access the services offered by the presentation agent. The services may include primitives such as "display data, accept user input, etc.".

The **object API** allows application processes to access the services of the object manager. The application process may run locally in the application interpreter or on a remote device. If the application is running on a remote device it needs to access the object API using an end-to-end protocol "application - distant object manager". The services may include primitives such as "prepare object, destroy object, etc.".

The **access API** allows the clients of the access agent (local application, object manager, presentation agent) to access the services offered by the access agent. The services may include primitives such as "get object, send object, etc.".

5.2 Service Support Function

As already mentioned above, the Service Support Function will be different depending on the specific multimedia service.

To give some examples the following categories have to be considered:

- Access Function (in case of retrieval and distribution services);
- Multipoint Control Unit (in case of conferencing services);
- Interworking Unit (between different multimedia services).

5.3 Host function

Figure 5 shows the "Host function" in more detail.

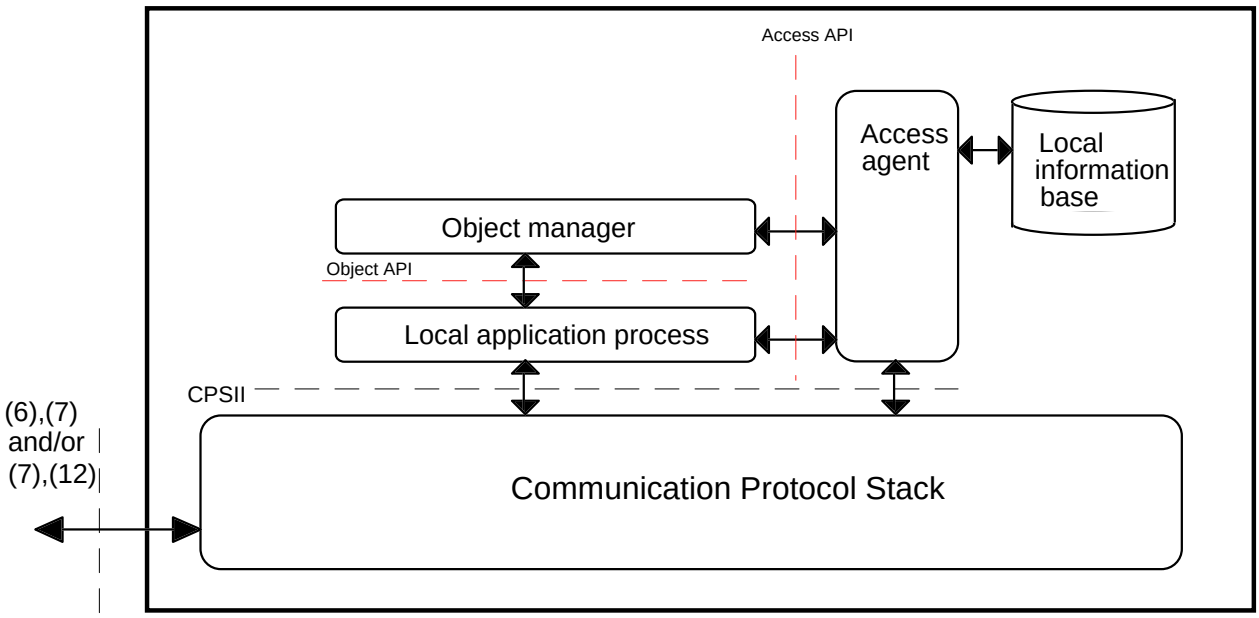


Figure 5: Host function

The description of the components within figure 5 correspond to the description under figure 4.

5.4 Database function

Figure 6 shows the "Database function" in more detail.

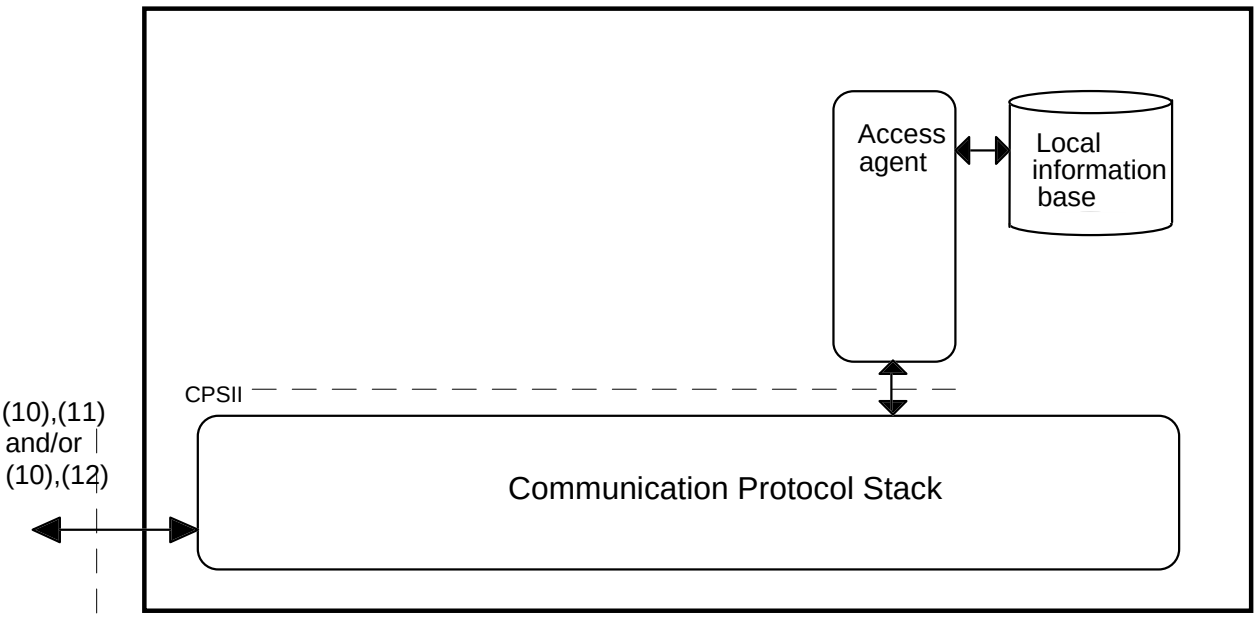


Figure 6: Database function

The description of the components within figure 6 correspond to the description under figure 4.

6 Interfaces/protocols between functional units

6.1 Man-Machine interfaces

In figure 1 the Man-Machine Interfaces (1) are shown between the "User" and the "Terminal Function".

Man-Machine Interfaces need to take into account "human factors" considerations as defined by ETSI TC HF and other committees, and are closely related to the specific multimedia application or service for which they are defined. Documents in this field are in the course of preparation within ETSI and should be considered as a basis as soon as they are available.

6.2 Network dependent protocols

In figure 1 the Network dependent protocols at the interfaces (2), (4), (5), (7), (9) and (10) are indicated.

It depends on the specific multimedia service which kind of networks are used and, therefore, which kind of protocols have to be implemented in the relevant Functional Unit.

6.3 End-to-end protocols

In figure 1 the end-to-end protocols at the interfaces (3), (6), (8), (11) and (12) are indicated.

It depends on the specific multimedia service which end-to-end protocols are used, i.e. which application platform is required.

Annex A: Examples of terminal architectures

A.1 Introduction

In the following examples possible terminal architectures are given, starting with the simplest one:

- Presentation terminal architecture;
- Multimedia information management terminal architecture;
- Script terminal architecture.

After that, two different possibilities for the realisation of the "Multimedia information management terminal architecture" are given.

A.2 Presentation terminal architecture

Compared with figure 4 of this ETR, only the following modules exists:

- Communication Protocol Stack (Network Termination, Transport Protocols);
- Local Application;
- Presentation Agent (Presentation Server).

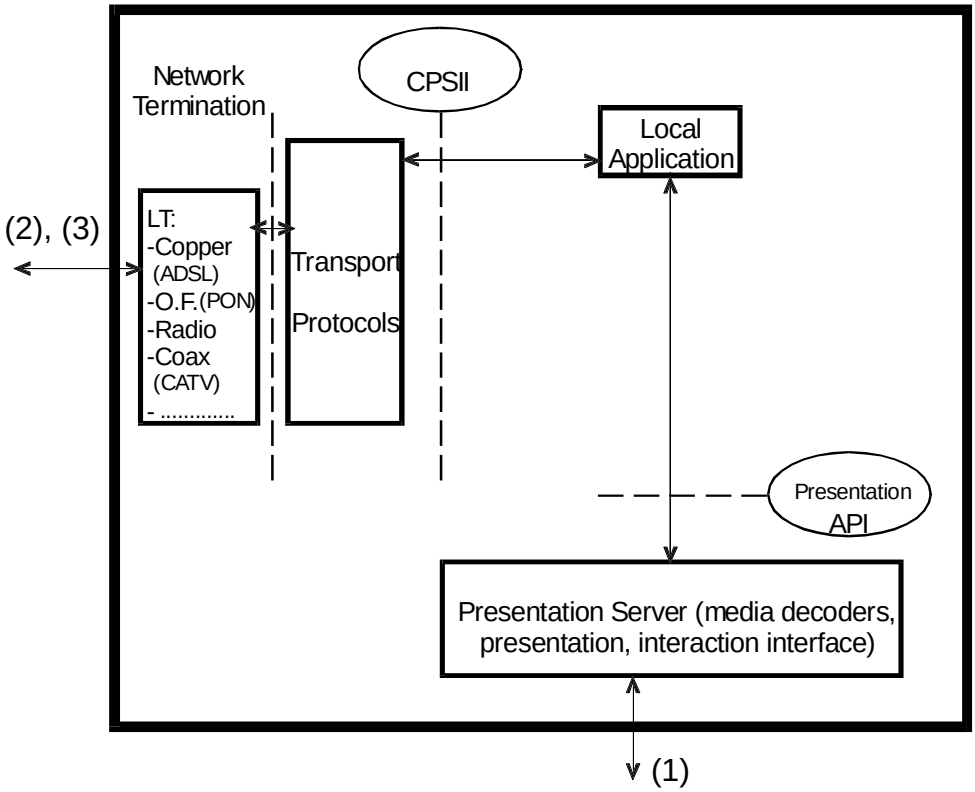


Figure A.1: Presentation terminal architecture

A.3 Multimedia information management terminal architecture

Compared with figure 4 in this ETR the following modules exist:

- Communication Protocol Stack (Network Termination, Transport Protocols);
- Local Application (without script facilities);
- Object manager Access agent and Local information base (Multimedia Information Management Module);
- Presentation agent (Presentation Server).

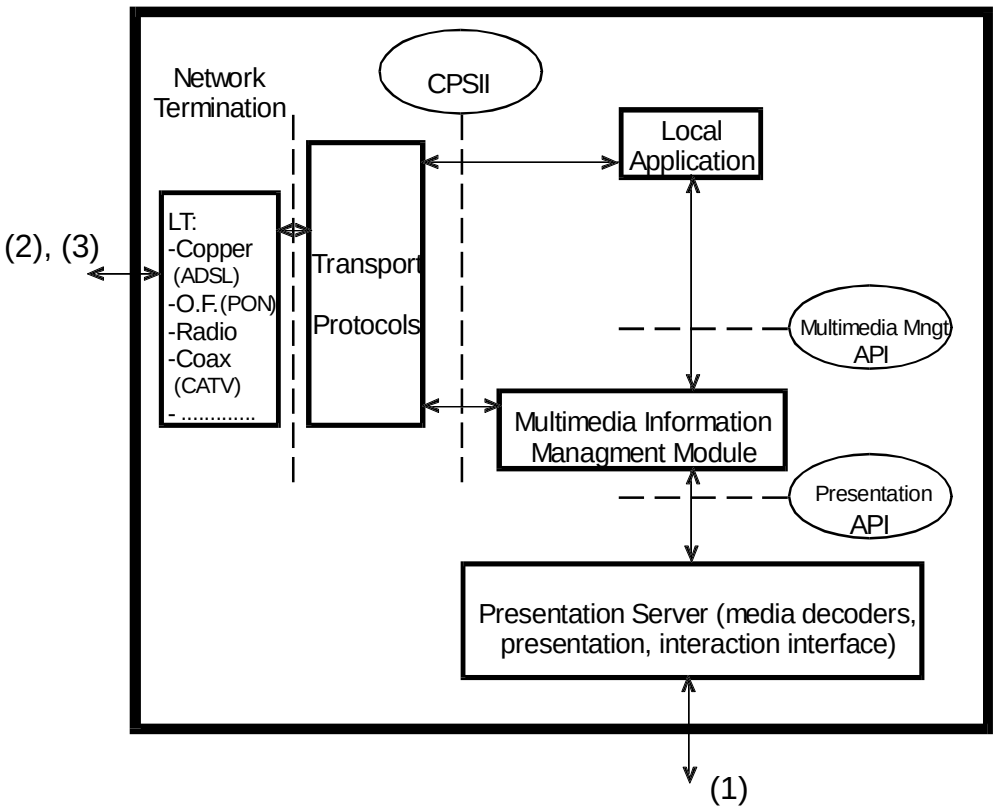


Figure A.2: Multimedia information management terminal architecture

A.4 Script terminal architecture (see figure A.3)

Compared with figure 4 of this ETR there exists all modules with the additional explicit mentioned "Script Interpreter".

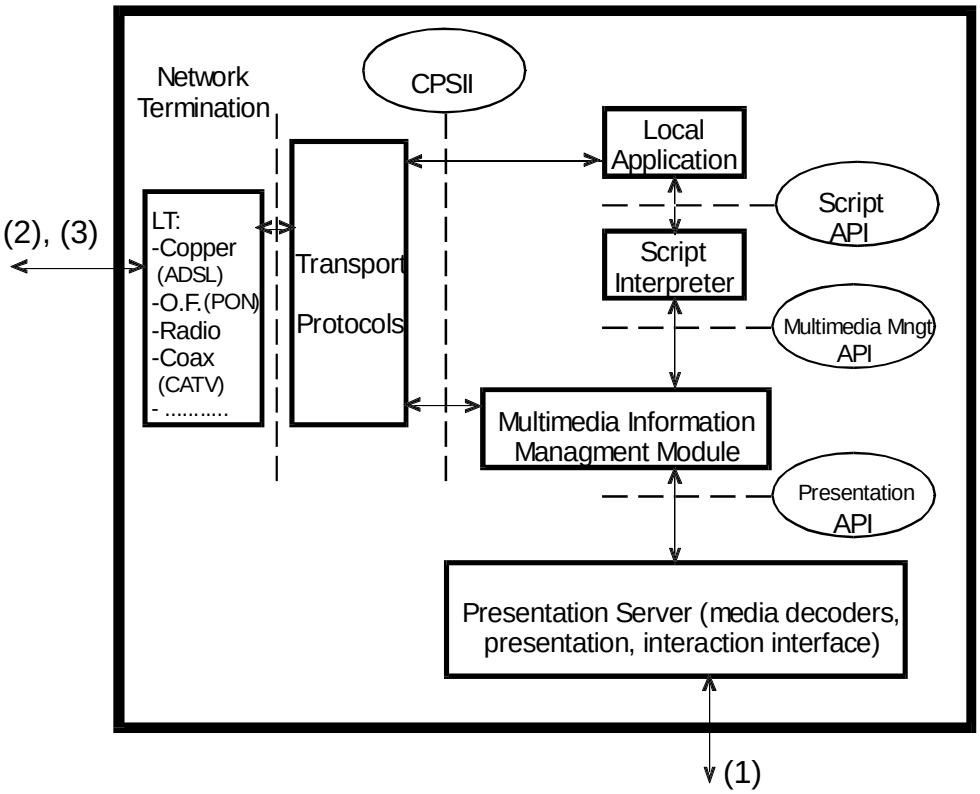


Figure A.3: Script terminal architecture

A.5 Videotex Enhanced Man-Machine Interface (VEMMI) example

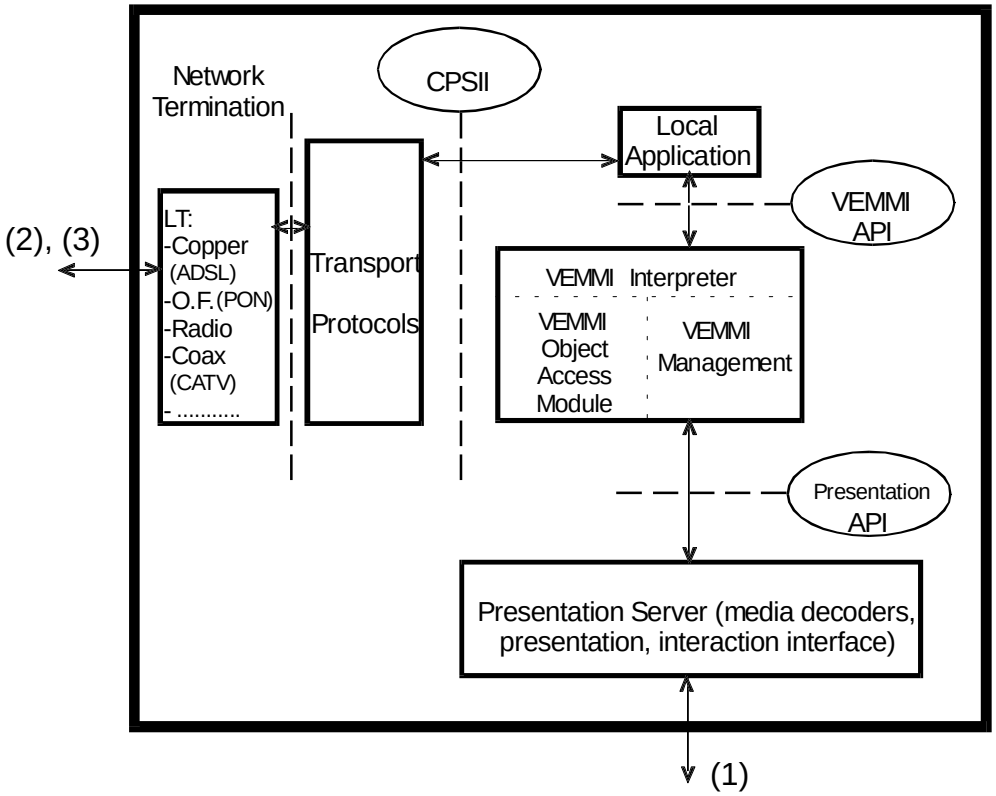


Figure A.4: Multimedia Information management terminal architecture (using VEMMI)

The VEMMI Interpreter acts as a parser for the VEMMI data received from the remote Server through the local application. A VEMMI access module controls the access to the object requested for the application (these objects can be stored in the Local Data Base or they have to be requested to the remote application). The requests can be referred to an object or to the content of the object in some specific cases. VEMMI Management module performs the rest of tasks related to the VEMMI protocol, local actions, objects behaviour, etc.

A.6 MHEG example

After the simple terminal we may have the Multimedia & Hypermedia (M&H) terminal dealing with the M&H level such as MHEG. This M&H terminal is represented in figure A.5 below:

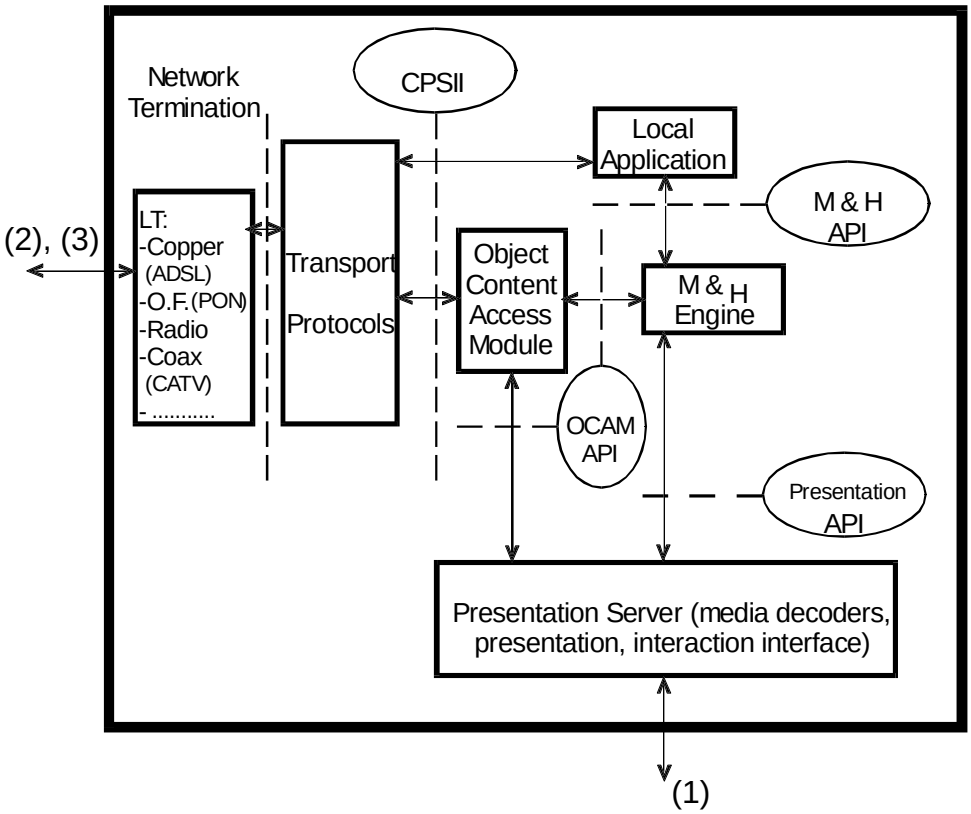


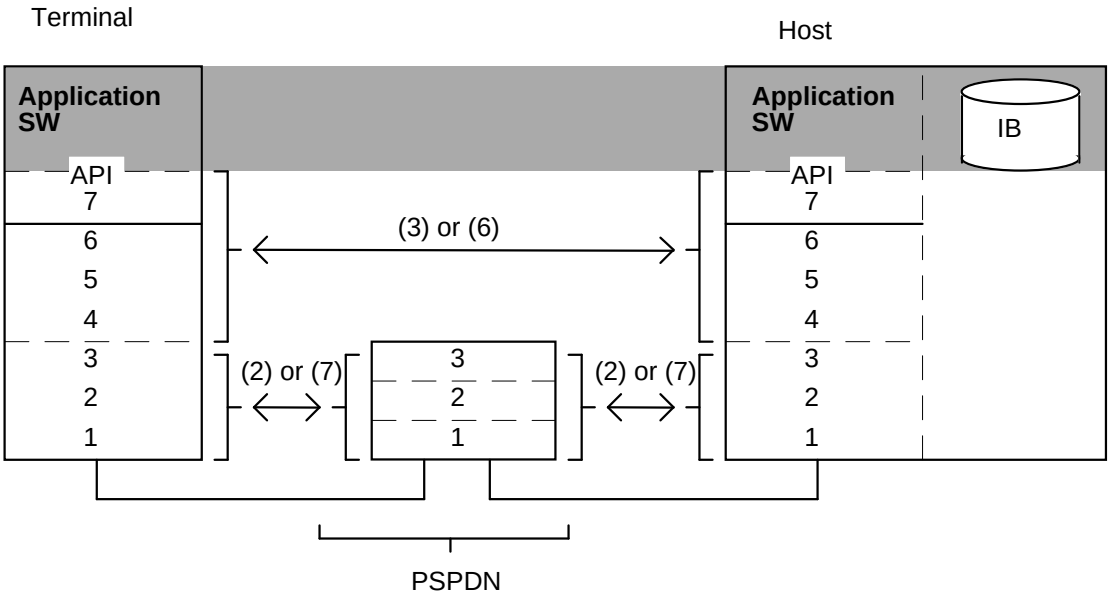
Figure A.5: Multimedia information management terminal architecture (using MHEG)

Compared with VEMMI, MHEG has in addition hyperlinks and time based synchronisation.

Annex B: Possible configurations

B.1 Example of a terminal-host configuration

Figure B.1 gives a simple terminal-host configuration as example in order to explain the area of this functional model.

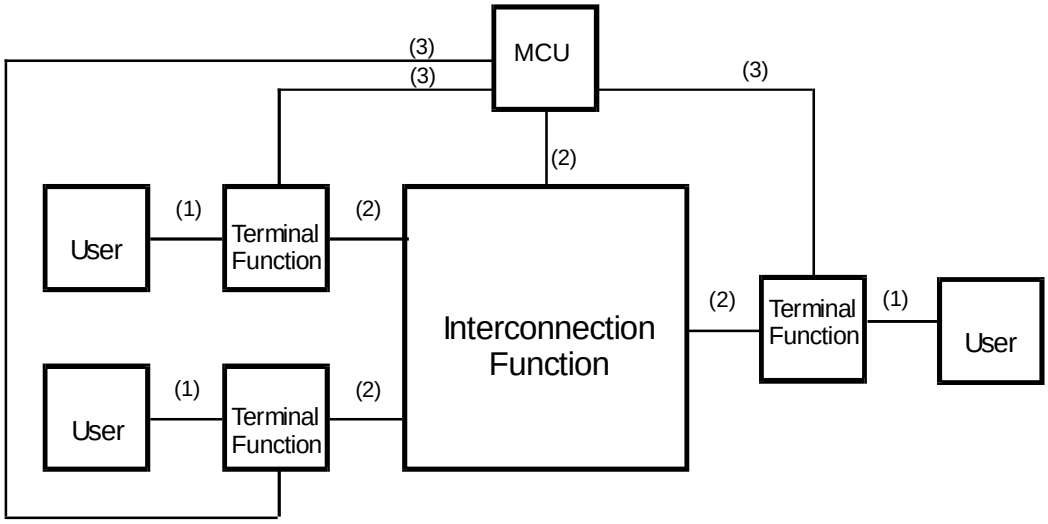


API: Application Programmable Interface
IB: Information Base
PSPDN: Packet Switched Public Data Network

Figure B.1: Terminal-host configuration

The functional model described in this ETR is applicable to the Application Software highlighted in figure B.1. In a communication environment an underlying protocol stack is necessary for the interconnection of the systems (e.g. terminal, host) via involved “relay systems” (e.g. PSPDN-nodes).

B.2 Example of conversational functional model - audiographic conference



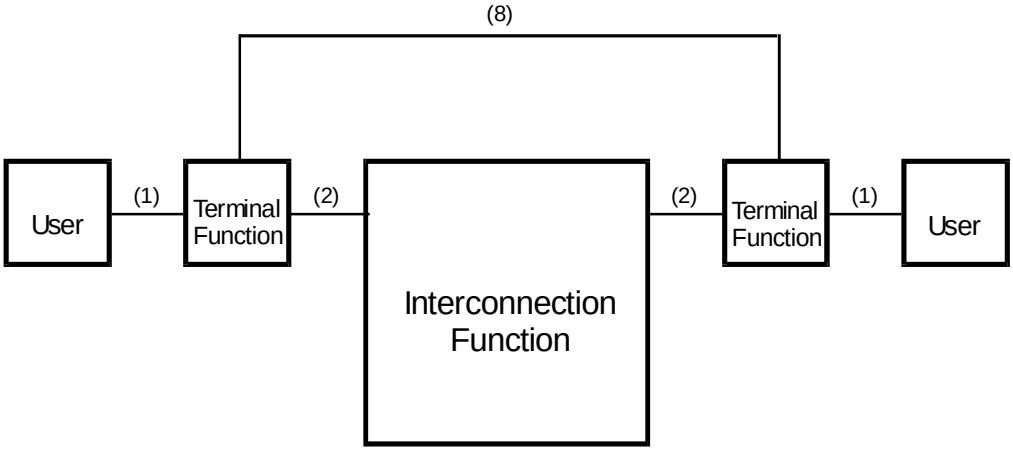
(1) = Man-Machine Interface.

(2) = Network dependent protocols which may differ depending on the type of network which provides access.

(3) = End-to-end protocols in relation to the audiographic application.

Figure B.2: Example of conversational functional model - audiographic conference

B.3 Example of conversational functional model



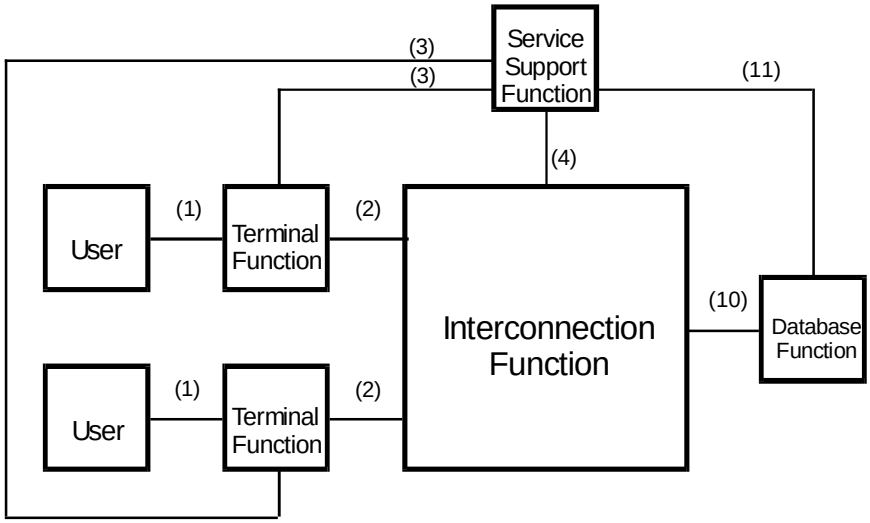
(1) = Man-Machine Interface.

(2) = Network Dependent Protocols which may differ depending on the type of network which provides access.

(8) = End-to-end protocols in relation to a conversational application.

Figure B.3: Example of conversational functional model

B.4 Example of distribution functional model



(1) = Man-Machine Interface.

(2) = Network Dependent Protocols which may differ depending on the type of network which provides access.

(3) = End-to-end protocols in relation to the Service Support Function.

(4) = Network dependent protocols which may differ depending on the type of network which provides access.

(10) = Network dependent protocols which may differ depending on the type of network which provides access.

(11) = End-to-end protocols in relation to the Database application.

Figure B.4: Example of distribution functional model

History

Document history	
March 1995	First Edition
February 1996	Converted into Adobe Acrobat Portable Document Format (PDF)

ISBN 2-7437-0104-8
Dépôt légal : Mars 1995