ETSI TECHNICAL REPORT

ETR 368

March 1997

Source: ETSI TC-TE

Reference: DTR/TE-02025bis

ICS: 33.020

Key words: PCI, taxonomy, terminal, testing

Terminal Equipment (TE); General architecture for Programmable Communication Interfaces (PCIs)

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 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Contents

Foreword 5

Introduction 5

1 Scope 7

2 References 7

3 Definitions and abbreviations 8
3.1 Definitions 8
3.2 Abbreviations 9

4 Overview and guidelines for readers 9

5 General model 10
5.1 Basic services 12
5.2 Benefits 13
5.3 PCI essentials 13

6 Exchange mechanism 14
6.1 General principles 14
6.1.1 The list resource procedure 15
6.1.2 The link procedure 15
6.1.3 The send procedure 15
6.1.4 The receive procedure 15
6.1.5 The unlink procedure 15
6.1.6 Client-server relationship 15
6.2 Special modes of operation 16
6.3 Resource management 16
6.4 The functions of the Exchange Mechanism 17
6.4.1 Generic structure of the Exchange Mechanism functions 17
6.4.2 The function GPCI_ListResources 17
6.4.3 The function GPCI_Link 19
6.4.4 The function GPCI_Unlink 20
6.4.5 The function GPCI_Put 20
6.4.6 The function GPCI_Get 21
6.4.7 The function GPCI_Poll 22
6.4.8 The function GPCI_SetSignal 23
6.4.9 Error handling 24
6.5 The Interface Data Units (IDUs) 25
6.5.1 IDUs of the user plane 26
6.5.2 IDUs of the Control Plane 26
6.5.3 IDUs of the administration plane 26

7 Interface characteristics 27
7.1 User and Provider description 27
7.1.1 Provider model 27
7.1.1.1 Communication Application (CA) 28
7.1.1.2 Provider Module (PM) 28
7.1.1.3 Provider structure 28
7.1.2 User model 29
7.2 Data flow through the interface 30
7.2.1 Memory management 31
7.2.2 File management 31
7.3 User/Provider relationship 31
7.3.1 One-to-one relationship 32

7.3.2 Many-to-one relationship 32
7.3.3 One-to-many relationship 33
7.3.4 Many-to-many relationship........ 34
7.3.5 Advise on GPCI_ListResources function........ 34
7.4 Co-ordination........ 35
7.4.1 Co-ordination handled by the User 35
7.4.2 Co-ordination handled by the Provider 38
7.4.3 Network addressing........ 40
7.5 Co-operation........ 41
7.6 Synchronization........ 42
7.7 Multiple interfaces 43
7.7.1 Hierarchical interfaces 43
7.7.1.1 Stack components configuration........ 43
7.7.1.2 Bypass facility........ 44
7.7.2 Parallel interfaces 44
7.7.3 Consequences of multiple interfaces system 44
7.8 Flow control........ 44
7.8.1 Outgoing flow control........ 45
7.8.2 Incoming flow control........ 46
7.8.3 Congestion........ 48

8 PCI checklist........ 48
8.1 The function GPCI_ListResources........ 49
8.2 The function GPCI_Unlink........ 50
8.3 The functions GPCI_Put and GPCI_Get........ 50
8.4 The functions GPCI_Poll........ 50
8.5 The functions GPCI_SetSignal 50
8.6 Error handling........ 50
8.7 User plane IDUs........ 50
8.8 Control Plane IDUs........ 50
8.9 Administration Plane IDUs 51
8.10 PCI checklist proformas 51

9 Testing aspects 54
9.1 Rules for PCI testing 54

10 Notation and structure of a PCI 55

Annex A: Clarification of the terms "API" and "PCI"........ 56

A.1 Categories of APIs........ 56

A.2 APIs for non-communication purposes 56

A.3 Programmable Communication Interfaces (PCIs)........ 57
A.3.1 Network related PCIs 57
A.3.2 Service related PCIs........ 57

Annex B: Samples of functional behaviour at the GPCI: a SDL description........ 58

Annex C: Bibliography........ 65

History........ 66

Foreword

This ETSI Technical Report (ETR) has been 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.

Introduction

Interfaces are specified in many different areas of information technology: access to databases, hardware devices or (human) end-user interfaces, etc.

Application Programmable Interface (API) is the generic term for a Programmable Interface which defines services offered at the border between two computer applications. Programmable Interfaces, enabling computer applications to use telecommunications services are called Programmable Communication Interfaces (PCIs).

PCIs form the basis for portability of applications across various hardware, heterogeneous communications and software platforms. A number of different PCIs are emerging from international, national or private standardization bodies. To gain efficiency, PCIs should have a common and homogeneous architecture, notation, structure and terminology.

This ETR describes, in an abstract manner, a global framework for the specification of telecommunications interfaces. The description makes no assumption about the implementation and the presentation of the manipulated data at/or through the interface.

Furthermore, this ETR:

- defines the terms used to describe the capabilities of communication systems which are in the domain of a telecommunications service provider;

- describes how to exchange information at a PCI level.

The main reason for providing this ETR is to give to "standards making bodies" basic material for consolidating their future PCIs. The generic model developed in this ETR allows different interfaces to be specified in a common way. Products implemented according to a PCI standard can be used in various telecommunications systems because they are independent of any operating system or any programming language.

Blank page

1 Scope

This ETSI Technical Report (ETR) specifies a general architecture for Programmable Communication Interfaces (PCIs). It describes the technical features and concepts in an abstract way to facilitate future interface specifications.

A communications system is composed of either software components or a combination of software and hardware components. A PCI enables software components of such a system to interact. Hardware interfaces and interfaces to (human) telecommunications users are not addressed in this ETR.

Where the architecture defined in this ETR is used in the development of future PCI standards, this ETR describes:

- the terminology to be used;
- the general model that provides the concepts of the architecture and an abstract model;
- a basic set of interactions and the general structure of the conveyed information;
- principles and general guidance for the design of telecommunications service providers;
- the basic aspects for testing a PCI;
- a guidance on formal notation and description.

This ETR should be seen as an overall architectural framework and guideline for future PCI specifications. The realization of this general architecture could be a set of consistent standardized PCIs which are in line with this ETR. Based on such standardized PCIs a set of consistent telecommunications products may be developed. Therefore, this ETR makes no assumption on the development of products but gives PCI designers:

- a general framework;
- a set of common guidelines;
- a collection of important and intrinsic aspects to be borne in mind for future PCI standards.

2 References

This ETR incorporates by dated and 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 publication referred to applies.

[1] ISO/IEC 10746-2: "Information technology - Open distributed processing Reference model; Part 2: Foundations".

[2] ITU-T Recommendation Z.100 (1993): "Functional specification and description language (SDL)".

[3] ITU-T Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".

[4] ISO/IEC 9646 (1991): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

[5] ITU-T Recommendation X.200 (1994): "Information technology - Open Systems Interconnection - Basic reference model".

3 Definitions and abbreviations

3.1 Definitions

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

administration Plane: A logical group of Interface Data Units (IDUs) related to administration and management aspects of the Provider.

Communication Application (CA): A component of the abstract entity User or the abstract entity Provider in a communications system, that uses and offers one or more telecommunications services.

connection: A temporary or permanent association between the User and a remote communications partner.

control Plane: A logical group of IDUs related to network control and signalling (outband only) of the Provider.

co-operation: A dynamic and temporary activation of a telecommunications resource in an existing connection.

co-ordination: The component of a communications system that handles the IDUs of the User Plane and the Control Plane to establish, control and release a physical and logical connection over an outband signalling network to a remote communication partner.

exchange mechanism: Specifies the basic set of Generic Programmable Communication Interface (GPCI) interactions and the generic structure of the information conveyed between User and Provider as a result of these interactions.

exchange mechanism function: One GPCI interaction.

Generic Programmable Communications Interface (GPCI): The interface at which the conveyance of information between the abstract entities, User and Provider, take place.

Interface Data Unit (IDU): Structured information exchanged between User and Provider. IDUs are classified into distinct, logical groups: IDUs of the Control Plane, IDUs of the User Plane, IDUs of the Administration Plane.

Local Application (LA): A component of the abstract entity User in a communications system, that uses one or more telecommunications service.

plane: Logical group of IDUs.

provider: The abstract entity, which offers one or more telecommunications services.

Provider Module (PM): A component of the abstract entity Provider in a communications system, that provides one or more telecommunications service and access to physical communications network(s).

synchronization: Information synchronization through multiple physical network channels.

system; communications system: The set of the abstract entity User, the interface GPCI, and abstract entity Provider interacting in a telecommunications environment to enable peer-to-peer communication.

telecommunications resource: A CA or PM.

User: The abstract entity of a communications system that uses one or more telecommunications services.

User Plane: Logical group of IDUs related to the peer-to-peer communication.

3.2 Abbreviations

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

API	Application Programmable Interface
ASN.1	Abstract Syntax Notation "number one"
ATS	Abstract Test Suite
CA	Communication Application
EBNF	Extended Backus Naur Form
EDI	Electronic Data Interchange
ETR	European Telecommunication Report
FDT	Formal Description Technique
GPCI	Generic Programmable Communication Interface
IDU	Interface Data Unit
ISDN	Integrated Services Digital Network
LA	Local Application
LAN	Local Area Network
NCO	Network Connection Object
OSI	Open Systems Interconnection
PCI	Programmable Communication Interface
PCI ICS	PCI Implementation Conformance Statement
PCI IXIT	PCI Implementation eXtra Information for Testing
PCO	Point of Control and Observation
PM	Provider Module
PSDN	Packet Switched Data Network
PSTN	Public Switched Telephone Network
RPC	Remote Procedure Call
SDL	Specification and Description Language
SUT	System Under Test
TE	Terminal Equipment
TTCN	Tree and Tabular Combined Notation

4 Overview and guidelines for readers

This ETR is structured in a sequence of clauses as described below.

Clause 5, "General model", specifies the components and basic services of a communications system. The benefits for both telecommunications users and telecommunications providers of standardized interfaces and the essentials of an interface, specified in line with this model, are described in this clause.

The interface of the general model, as seen from a telecommunications user, is described in clause 6, "Exchange mechanism". All mandatory, optional and conditional characteristics of the functions used to convey information and the structure of the information are defined. The differences between information, which are a local matter, and which are related to the peer-to-peer communication, and those which are physical network related are also explained.

The interface of the general model, as seen from a telecommunications provider, is described in clause 7, "Interface characteristics". It provides a description of the intrinsic characteristics and capabilities of a communications system. Physical communications network related aspects, synchronization, co-ordination and co-operation aspects of a telecommunications provider are described.

All aspects which are outside the scope of the interface of the general model but are important and necessary for a standardized interface following the architectural framework of this ETR are collected in clause 8, "PCI checklist".

Some general remarks concerning the testing aspects are described in clause 9, "Testing aspects", whereas clause 10, "Notation and structure of a PCI", concerns to the general requirements for a formal description which should be used in an interface specification.

5 General model

The architectural framework and guideline for the specification of programmable interfaces in a communication system is based on the description of the structural model presented in this clause.

The model does not make any assumption about the implementation of interfaces and of the actual representation of data manipulated at, or through, the interface. The model identifies the relevant characteristics and suppresses irrelevant detail to be simple and generic. It is independent of operating systems, programming languages and hardware platforms.

This model establishes the framework for the specification of interfaces in communications systems, which are referred to as open systems. Such systems are composed of one or more computers, associated software and hardware, physical communication network(s), external equipment (e.g. terminals) and human operators. Interfaces in such a system may be located at different levels in the hierarchy of standardized layers. The concepts and principles developed here allow the definition of interfaces enabling the interconnection and exchange of information. This ETR addresses interfaces which are:

- application layer oriented;
- network layer oriented;
- oriented towards standardized layers, situated between network and application layer.

Standardized layers means any layer of the Open Systems Interconnection (OSI) model (see ITU-T Recommendation X.200 [5]) as well as protocols standardized by international, national, bilateral or private standardization bodies. Interfaces to human beings (e.g. human operator) and interfaces to physical networks are not addressed in this model.

A communications system is comprised of a set of components located on the same physical medium (computer) or on distributed physical media (different, connected computers). The structure of such a system (the architecture) identifies the interrelationships among its components. These components exchange information among each other. Exchange of information is performed by a set of interactions.

Subsequently, abstract definitions of such components are called abstract entities. According to ISO/IEC 10746-2 [1] these entities are to be seen as objects. Communication is, therefore, the transmission of information between objects as a result of interactions (see also ISO/IEC 10746-2 [1]).

Firstly, two basic entities are identified in the simple generic model of this ETR:

- User;
- Provider.

The Provider is the entity which offers telecommunication service(s). The User is the entity which uses the offered service(s). Different products may, as a realization of the Provider, offer their services in many different ways. In order to be abstract from such individual properties a mechanism is needed to make User and Provider independent from each other. A generation of such a mechanism should be specified by the Generic Programmable Communication Interface (GPCI).

Figure 1 illustrates a communications system represented with the entities explained above.

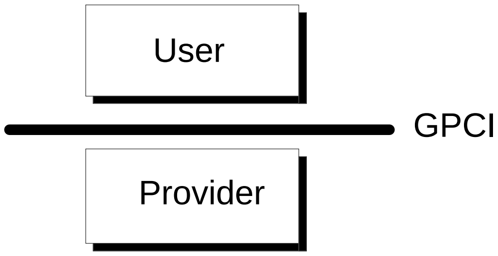


Figure 1: Simple generic model

The communication between User and Provider is the conveyance of information as a result of one or more interactions. The standardized means for such interactions should be provided by PCIs according to the GPCI as defined in this clause.

Each entity of the simple generic model may itself be built by a set of components. Refining the User and the Provider leads to the introduction of new entities:

- Local Application (LA), as a component of the User;
- Communication Application (CA), as a component of the User or the Provider;
- Provider Module (PM), as a component of the Provider.

According to ISO/IEC 10746-2 [1], the distinct entities LA, CA, and PM are also to be seen as objects.

The LA is the entity which uses telecommunications service(s). It provides access to these services for other components of a system which are outside the scope of the generic model. The CA is the entity which uses and offers telecommunications service(s). The PM entity offers telecommunications service(s) and provides access to a physical communications network of the system.

The entity pairs:

- LA and CA;
- LA and PM;
- CA and CA;
- CA and PM,

exchange information as a result of interactions which take place at a PCI according to the GPCI. A User is the composition of one or more LA and none, one or more CA, whereas the Provider is a composition of none, one or more CA and one or more PM. Therefore, a communications system consistent with the simple generic model is composed of at least one LA and one PM.

This refinement of User and Provider may lead to a very complex architecture of a system with multiple PCIs, which are in line with the GPCI. If multiple PCIs are in the same system the structural relationship between these PCIs is:

a) hierarchical: one or more interfaces between different layers of the stack in a system;
b) parallel: different interfaces on the same hierarchical level.

A system may encompass any composition of hierarchical and parallel interfaces.

Different interfaces of a system are independent entities. Consequently, knowledge of the existence or functionality among the PCIs in a system is neither needed nor available.

Different components of the Provider offering the same telecommunications service(s), which are specified according to the GPCI model, should be behaviourally compatible with respect to a fixed set of criteria. This means that one component can replace another component of the Provider without observable change to the User.

Principles for the design of Providers adhering to the conceptual framework of this ETR are described in clause 7.

5.1 Basic services

An entity is characterized by its behaviour and its state. Interactions may change the state of an entity. Since User and Provider are independent entities, knowledge about the internal state of each other is neither needed nor available.

The first externally observable state is the result of an initial process which makes User and Provider ready for intercommunication. This process is called binding (see also ISO/IEC 10746-2 [1]), resulting in a local link being established between User and Provider.

A link can be characterized as:

- permanent: User and Provider are linked as long as the system is operable; data can be exchanged at any time;

- temporary: User and Provider are linked only for a certain period of time.

The User may establish one or more local links to a Provider, i.e. the components of a User may establish one or more links to the components of a Provider. The relationship between User and Provider concerning the number of established links is characterized as follows:

a) one-to-one relationship: one component of the User is linked to one component of the Provider at one time;

b) one-to-many relationship: one component of the User is linked to more than one component of the Provider at the same time;

c) many-to-one relationship: more than one components of the User are linked to one component of the Provider at the same time;

d) many-to-many relationship: more than one components of the User are linked to more than one components of the Provider at the same time.

Principles for, and consequences from, the design of Providers concerning the ability to handle more than one link are described in clause 7.

After the binding process, User and Provider may exchange information. The conveyed information has a defined syntactical structure. This structure depends on a fixed set of criteria, defining the specific characteristics of a PCI. It includes administrative data (e.g. a unique link identification) and the Interface Data Units (IDUs). IDUs are categorized in distinct, logical groups by their use and purpose. These logical groups are called Planes:

- the User Plane: logical group of IDUs related to the peer-to-peer communication;

- the Control Plane: logical group of IDUs related to network control and signalling, which is applicable only to outband signalling (e.g. supplementary services);

- the Administration Plane: logical group of IDUs related to administration and management aspects of the Provider; IDUs of this group are related to the management of the Provider (configuration of the Provider, traces, etc.), bridges to tools (e.g. to printers) and the mechanism to handle exceptional situations in the Provider (e.g. congestion).

The mode of operation supported at the GPCI is characterized by the behavioural roles of User and Provider. User and Provider may interact in a client-server relationship. After establishment of a local link, the client entity (the User) requests services which are performed by the server entity (the Provider).

Telecommunications service(s) offered by the Provider of a system may require that the information is exchanged without any manipulation at or through the PCI to increase traffic performance. Special modes of operation become necessary:

a) transparent mode: after establishment of a local link, the exchange of information between User and Provider may be performed with special functions which makes the PCI become transparent;

b) bypass mode: an external equipment (e.g. a telephone) is physically connected to the Provider; intercommunication between this external equipment and the Provider is not performed with the interactions which are part of this ETR. As a result, the PCI is bypassed; the User may or may not initiate and terminate the process of bypassing the PCI; the external equipment and the Provider may request and perform services in parallel (interactive) or sequentially.

The process, which finishes the intercommunication, is called unbinding (see also ISO/IEC 10746-2 [1]), resulting in the local link between User and Provider being terminated, thus User and Provider are unlinked.

The interactions supported at the GPCI and the general structure of the information conveyed with these interactions, are specified in clause 6 "Exchange Mechanism" as well as general rules concerning intrinsic aspects of communication between User and Provider.

5.2 Benefits

Interfaces in the telecommunications area are already specified by international, bilateral, national or private standardization bodies. Presently, all these specifications differ in terminology, notation and structure. Many organizations are currently developing interface specifications with the same deficiencies. To abolish these deficiencies, this ETR provides an architectural framework and guideline for the designers of future interfaces.

This general architecture is specified independently from operating systems, programming languages and hardware platforms. Therefore, benefits for both telecommunications service users and telecommunications service providers are identified:

a) interoperability: components developed by different manufacturers, which are conformant with a standardized PCI, work well together if they are used in the same system;

b) compatibility: components, which offer the same service(s) and which are developed in conformance with a standardized PCI, are compatible in their behaviour; they can be exchanged by each other without reflecting the other components of the system;

c) reusability: components developed in conformance with a standardized PCI, which are designed to provide a particular service and can be used repetitively.

A PCI specified in line with the GPCI should help customers to identify a PCI product satisfying their needs and assists manufacturers to develop a conforming product.

5.3 PCI essentials

The model designed in this ETR is generic and identifies the relevant characteristics for a class of PCIs. The complete specification of a PCI, which is in line with the GPCI, should capture the interactions that can be performed at the interface, the observable states of the communicating objects (the components of a system) affected by the interactions and the circumstances in which the operations are appropriate. The PCI specification should be designed independently from any realization and should use a formal description technique to maximize clarity and minimize ambiguity, inconsistency and incompleteness.

A PCI specification should consist of:

a) the specification of the context within which the accessible (telecommunications) services operate;

b) the specification of the behaviour observable at the interface.

A PCI specification, modelled in line with the GPCI, should define:

- the scope of the PCI: description of the need for and the purpose of this PCI;

- the field of application of the PCI: description of the concerned telecommunications area, limitations and restrictions, if there are any;

- the infrastructure supported by the PCI: description of the infrastructure of the communications system of this PCI (e.g. one computer or distributed computers; local and/or remote PCIs);

- the architecture of the system: specification of the components of the system and their interrelationships;

- location of the PCI: specification of the location of the PCI in the hierarchy of standardized layers in a communications system;

- services of the PCI: specification of the provided telecommunications, administration and management services, as well as characterization of the accessible physical network(s); specification of restrictions in the use of the provided services, if there are any; the definition of profiles or classes for a PCI, if there are any.

6 Exchange mechanism

As introduced in clause 5, User and Provider exchange information by interactions which take place at the GPCI. These interactions, which are subsequently called functions, are described independently from operating systems, programming languages, and hardware platforms.

A PCI, which is specified according to the GPCI, should provide the basic set of GPCI functions as specified in this clause. A PCI may detect specific needs so that other functions not specified in this ETR become necessary. The definition of additional functions is allowed in a PCI specification if they are a real extension to the defined GPCI functions (no redundancy).

The structure of the information exchanged between User and Provider is described without making any assumption of the actual representation of the data manipulated at or through, a PCI. The information exchanged between User and Provider appears as a parameter list of the used function. A parameter, if specified by the user, is called an input parameter. A parameter is called output parameter, if specified by the Provider. Transient parameters (bi-directional), which may be specified by the User and the Provider, are called input-output parameters. The parameters are characterized as being mandatory, optional or conditional.

Each standardized PCI, which is specified according to the GPCI, should define the structure of the information, the data types of the parameters, the encoding rules and should specify the dynamic behaviour at that PCI with a formal specification language such as the ITU-T Specification and Description Language (SDL) (see ITU-T Recommendation Z.100 [2]) or any other Formal Description Technique (FDT).

Each function returns a "so called" completion code, which is an output parameter. Subclause 6.4.9 describes general rules for handling conflict situations and for the definition of error codes.

General aspects concerning the administration and management of resources involved in a communications system are described in subclause 6.3.

The telecommunications service(s) offered by the Provider of a system may require that the information is exchanged without manipulation at or through the PCI, to increase traffic performance. General aspects of this requirement are described in subclause 6.2.

A PCI specified in line with the GPCI should give more details on operating system dependencies, programming language dependencies (language bindings) and hardware platform dependencies. These details should be well isolated from the PCI specification itself (e.g. in informative annexes or separate clauses).

This ETR provides a "generic Exchange Mechanism". It should be used as a guideline giving general rules and recommendations for the specification of a PCI, which is in line with the GPCI.

6.1 General principles

Before specifying the set of basic GPCI functions and the structure of information conveyed with them, this subclause describes their general use and purpose.

6.1.1 The list resource procedure

The Provider is a black box from the viewpoint of the User. As described in clause 5, the Provider may be composed of a set of CAs and PMs. Each component, subsequently called telecommunications resource, offers one or more telecommunications service.

If the Provider is composed of more than one telecommunications resource, the User should have the possibility of knowing:

a) which telecommunications resources are offered in the Provider;
b) to which physical network(s) a telecommunications resource enables access.

To select one of them for own use, the User should get a list as a result of the function GPCI_ListResource, with all necessary information characterizing the telecommunications resources of the Provider (see subclause 6.4.2).

6.1.2 The link procedure

User and Provider can communicate only if an end-to-end path between them is established. Therefore, the successful establishment of a local link between User and Provider indicates that both entities are ready for communication. A local link is established as a result of the function GPCI_Link (see subclause 6.4.3).

6.1.3 The send procedure

After establishment of a local link to the Provider, the User may send information using the function GPCI_Put (see subclause 6.4.5); as a result of this function, the information is conveyed from the User to the Provider.

6.1.4 The receive procedure

After establishment of a local link, the User may receive information using the function GPCI_Get (see subclause 6.4.6); as a result of this function the information is conveyed from the Provider to the User.

6.1.5 The unlink procedure

When there is no longer a need for interaction, the terminating procedure should be performed. A local link is terminated as result of the function GPCI_Unlink (see subclause 6.4.4).

An established local link reserves telecommunications resource(s) of the Provider which are thus not available to other components of the User. Therefore, the local link should be released as soon as possible.

6.1.6 Client-server relationship

In a client-server relationship, the Provider should have the possibility to:

- announce incoming peer events;
- announce exceptional or erroneous situations.

The intercommunication between User and Provider should be initiated by the User. Therefore, the User is responsible for:

- transmission of information;
- reception of information.

Since the User cannot know the time at which information is available, at least one of the following mechanisms should be applied to align the independent entities, User and Provider:

1) synchronous: the User uses the function GPCI_Get and waits until information arrives;

2) asynchronous: two different mechanisms are possible:

   a) the poll mechanism: the User should ask the Provider, if any information is available using the function GPCI_Poll (see subclause 6.4.7); a list of available information is returned as a result of this function; if a single information is available, the User should perform the receive procedure; if n (n > 1) information are available, the User should perform the receive procedure n times; if no information is available, the User may poll again after a period of time; the GPCI specification does not constrain the User with respect to the length of the time period between successive polling actions;

   b) the interrupt mechanism: the User should specify a call back routine (an interrupt routine) using the function GPCI_SetSignal (see also subclause 6.4.8); as a result of this function the interrupt routine should be called automatically each time, when the Provider needs to convey information to the User; if the call back routine is invoked, the User should perform the receive procedure.

6.2 Special modes of operation

Telecommunications service(s) offered by the Provider of a system may require that the information is exchanged without manipulation at or through the PCI, to increase traffic performance. If such real-time oriented services are offered in a communications system, continuous data flow is necessary. The GPCI becomes transparent or is bypassed (see also clause 5).

With regard to the link between User and Provider, real-time oriented data exchange implies:

a) for a temporary link: the User should perform the link procedure as described; after the establishment of the local link the User should perform the sending procedure to inform the Provider, that continuous data flow is intended; the User and the established local link should not be involved in the data transmission until the User releases the local link (on request or self initiated) with the unlink procedure; only a subset of the GPCI functions are used;

b) for a permanent link: the Provider and a component outside the scope of the GPCI model (e.g. an external equipment) are physically connected permanently; neither the User (or a component of the User) nor the GPCI are involved in this communication; no GPCI function is used.

These special modes of operation for real-time oriented data exchange become very important for multimedia systems which are outside the scope of this ETR.

6.3 Resource management

User and Provider exchange information by using resources of their communications system. Necessary resources are:

- memory resources (storage medium);
- telecommunications resources of the Provider;
- (physical) network channels;
- local link resources.

A PCI specification which is in line with the GPCI specification should define the resource administration and management of the communications system.

It should define:

- used memory resources (file, buffer);

- proper or shared memory resources for User and Provider;

- which entity provides memory resources for the exchange of information and which entity releases these dynamically allocated memory resources;

- the limitations and restrictions concerning the telecommunications resources of the Provider, if any (e.g. a resource may be able to handle only a fixed number of local links);
- the limitations and restrictions concerning the physical network(s) accessible through the Provider, if any (e.g. an ITU-T Recommendation X.25 (see annex C) communication board may limit the number of physical connections to a peer communications partner at a time).

6.4 The functions of the Exchange Mechanism

The GPCI interactions described on a detailed level including the generic structure of the exchanged information, are specified in the following subclauses.

6.4.1 Generic structure of the Exchange Mechanism functions

The information to be exchanged with a function is composed of a list of parameters. The GPCI function calls are presented according to the following syntax:

GPCI_FunctionName (parameter 1, parameter 2, ..., parameter n).

This syntax specified using the Extended Backus-Naur Form (EBNF) is:

-- BEGIN

ExchangeMechanismFunction	::=	"GPCI_" FunctionName "(" Information ")"
FunctionName	::=	String
Information	::=	ParameterList | empty
ParameterList	::=	Parameter { "," Parameter }
Parameter	::=	InputParameter | OutputParameter
InputParameter	::=	Value | Reference
OutputParameter	::=	Value | Reference
Value	::=	data
Reference	::=	reference to allocated memory resource containing the data

-- END

A PCI specification should replace the preceding letters GPCI_ by a unique identification of that PCI.

6.4.2 The function GPCI_ListResources

The Provider may consist of more than one resource providing the same telecommunications services. The structure of a Provider is invisible to the User. Therefore, a User should have the possibility to get all characteristics and capabilities of each, or a specific kind of, telecommunications resource of the Provider.

This service is supported by the GPCI_ListResources function:

GPCI_ListResources (*UserId, QueryMask, ResourceList, Result*)

If the Provider is composed of only one telecommunications resource which offers one telecommunications service and access to one physical network, this function may not be provided (optional). The designer of such a PCI should notice that the evolution of a "single resource Provider" to a "many resource Provider" is only possible by specifying a new PCI standard. This may lead to non-conformity between these standards.

As a result of the function GPCI_ListResources, a list of telecommunications resources is put at the User's disposal. A PCI which is specified according to the GPCI should define if the result of this function is the list of currently available or of existing resources of the Provider.

The parameter UserId (conditional):

The User identifies itself for authorization and security aspects with a system-wide known unique identification with the parameter *UserId* (input parameter). If the User consists of only one component, the use of this parameter is optional, otherwise it should be mandatory.

The parameter QueryMask (conditional):

The User should be able to decide whether it wants to get information about all or a specific type of telecommunications resource. With the parameter *QueryMask* (input parameter) the User specifies the type of resource of the Provider it is looking for. The use of this parameter is optional if the result of the GPCI_ListResources function is the list of all existing telecommunications resources. It becomes mandatory if the list contains all actually available telecommunications resources of the Provider.

The parameter ResourceList (mandatory):

In the case of successful completion of the function (see parameter *Result*), a list of telecommunications resource(s) depending on the parameter *QueryMask* should be put at the User's disposal in the parameter *ResourceList* (output parameter). The parameter *ResourceList* is a complex data structure containing the number of telecommunications resources and the characteristics and capabilities of every single resource:

1) NumberOfResources: number of telecommunications resources;

2) ResourceRecord: characteristics and capabilities of one telecommunications resource.

A *ResourceRecord* consists of a set of fields, describing the characteristics and capabilities of a Provider's telecommunications resource. The information of each available resource is dependent on:

- the scope of a PCI;
- the location of a PCI in the communications system;
- the telecommunications service(s) supported by the Provider of a PCI.

A PCI specification which is specified according to the GPCI should define the structure of a record, the meaning and the data type of each field of this record and the length of a single record.

The following fields should at least, be in a *ResourceRecord*:

1) ResourceId: unique identification of the available resource;

2) ServiceId: unique identification of the telecommunications service provided by the resource;

3) NetworkId: unique identification of the physical network;

4) NetworkControl: access to network signalling and control supported;

5) AdditionalItem: used for specification of characteristics and capabilities of a telecommunications resource, not covered by the above fields; a variable number of AdditionalItems may be specified.

The fields of a record should give the User a technical overview of the characteristics, the capabilities and the services provided by a telecommunications resource, which are important from the viewpoint of a User. A PCI which is specified according to the GPCI should define:

a) name and data type of the *AdditionalItems;*
b) which field of a *ResourceRecord* could be used to specify the value of the parameter *QueryMask*.

The parameter Result (mandatory):

The value of the parameter *Result* indicates success if the *ListResources* function was accepted. On successful completion, none, one or more telecommunications resource records should be conveyed to the User.

If the memory resource referenced for the parameter *ResourceList* is insufficient, only the number of available telecommunications resources (see *NumberOfResources*) may be provided. In this error case, the value of the parameter *Result* should indicate failure (e.g. not_enough_memory).

In any other error case, the *ResourceList* should be empty and the value of the parameter *Result* indicates the reason for failure with an error code.

6.4.3 The function GPCI_Link

No interaction between User and Provider should be possible before the establishment of a local link. Thus, the User should be registered before being able to interchange information with the Provider. This service is supported by the GPCI_Link function:

GPCI_Link (*UserId, ResourceId, NetControlAccess, LinkId, Result*)

The parameter UserId (conditional):

If the User is composed of more than one component, this parameter identifies the component which is the owner of the requested local link (input parameter). If the User consists of only one component, the use of this parameter is optional, otherwise it should be mandatory.

The parameter ResourceId (conditional):

The User should provide the value of the parameter *ResourceId* (input parameter) to address the telecommunications resource it wants to access. This unique resource identification is one of the *ResourceId*s of the *ResourceRecord* (see subclause 6.4.2). If the Provider is composed of only one telecommunications resource which offers one telecommunications service and access to one physical network, the use of this parameter is optional, otherwise it should be mandatory.

The parameter NetControlAccess (conditional):

If the selected telecommunications resource, identified in the parameter *ResourceId*, supports *NetworkControl* (see subclause 6.4.2) the User should determine if it wants to handle network control and signalling or not. If the selected resource does not support *NetworkControl* the parameter *NetControlAccess* (input parameter) should not be used or if specified, be ignored by the Provider. The use of this parameter is optional if access only to physical network(s) with inband signalling is supported, otherwise it should be mandatory.

A PCI in line with the GPCI should specify the meaning and consequences of enabling network control and signalling (all signalling and control events are passed to the User, a part of them, etc.).

The parameter LinkId (conditional):

If the function is executed successfully, the value of the parameter *LinkId* (output parameter) is the unique identification of the established local link. This identification should be used in all successive GPCI functions.

A User may establish more than one local link to one telecommunications resource or establish more than one local link to different telecommunications resources. In any case, the User should be able to handle all its established local links identified by the *LinkId*s. If only one link can be established at a time the use of this parameter is optional, otherwise it should be mandatory.

The parameter Result (mandatory):

The value of the parameter *Result* indicates whether or not the GPCI_Link function was accepted. On successful completion, a local link to one telecommunications resource has been established. A Provider may reject the request to establish a local link. In these cases the value of the parameter *Result* indicates the reason for failure with an appropriate error code (e.g. no-telecommunications-resource-available).

6.4.4 The function GPCI_Unlink

The GPCI_Unlink function is issued by the User to release a local link to the Provider. No further exchange of information between User and Provider should be possible. The end-to-end path between User and Provider is terminated.

GPCI_Unlink (*UserId, LinkId, Result*)

The parameter UserId (conditional):

Identification of the component of the User, who is the owner of the local link (input parameter). The value of this parameter is the same as of the parameter *UserId* of the function GPCI_Link. If the User consists of only one component, the use of this parameter is optional, otherwise it should be mandatory.

The parameter LinkId (conditional):

Unique identification of the local link which the User (or a component of the User) wants to release (input parameter). The User should be the owner of that local link. The value of this parameter is the same as of the parameter *LinkId* of the function GPCI_Link. If only one link can be established at a time, the use of this parameter is optional, otherwise it should be mandatory.

The parameter Result (mandatory):

The value of the parameter *Result* indicates if the GPCI_Unlink function was accepted. On successful completion, the local link to a telecommunications resource of the Provider is terminated. Otherwise the value of the parameter *Result* indicates the reason for failure with an appropriate error code. The communication path between User and Provider is not released.

If the Provider cannot accept the GPCI_Unlink, he should be able to reject the termination of the local link and the User should be able to handle this rejection. Each PCI specification in line with this ETR should describe the situations and solutions of those conflicts.

6.4.5 The function GPCI_Put

After establishment of the local link between User and Provider, the User may call the GPCI_Put function to send information:

GPCI_Put (*LinkId, IDU, Result*)

The parameter LinkId (conditional):

The value of the parameter *LinkId* (input parameter) is the unique identification of the established link between User and Provider. The value of this parameter is the same as of the parameter *LinkId* (see subclause 6.4.3). If only one link can be established at a time, the use of this parameter is optional, otherwise it should be mandatory.

The parameter IDU (mandatory):

The IDUs exchanged with the function GPCI_Put belong to one of the three planes. The parameter *IDU* (input parameter) of the GPCI_Put function is a complex data structure. It is composed of n *IDUItem* (n > 0). The meaning and data structure of the parameter *IDU* should be specified by the designer of a standardized PCI depending on:

- the scope of that PCI;
- the location of that PCI in the communications system;
- the service(s) offered by the Provider;
- the accessible physical network(s).

Therefore, as an example, *IDUItem*s building the IDU may be:

a) IDUId: unique identification of the IDU; predefined by the designer of a standardized PCI;

b) ConnectionId: the User may use the local link to the Provider to establish one or more than one network connections to remote communications partner(s). Therefore, for each network connection a unique identification should be provided. The value of the parameter *ConnectionId* (input parameter) identifies, without ambiguity, the connection on which an IDU is to be transmitted. If the item *ConnectionId* is used, a PCI specification should define, for the phase of connection establishment to a peer communications partner, who generates and maintains the *ConnectionIds*, the User and/or the Provider. Special attention should be paid to the case, where a remote communications partner establishes the network connection (incoming call);

c) IDUNumber: in addition to the parameter *ConnectionId* another unique identification may be necessary, to address an IDU in exceptional situations inside the Provider (for further information see also subclause 6.5.3). The value of the parameter *IDUNumber* (input parameter) provides this information. The parameter *IDUNumber* should be used for flow control aspects at the local link (e.g. the Provider announces network congestion, invalid data for the current state of the telecommunications resource, etc.).

A PCI specification, which is in line with this ETR, should define the meaning and the data structure of the *IDU*s and the data type of each *IDUItem*. The ASN.1 description (see ITU-T Recommendation X.208 [3]) or any other FDT may be used.

The parameter Result (mandatory):

The value of the parameter *Result* indicates if the GPCI_Put function was accepted. On successful completion the information should be conveyed to the Provider. Otherwise the value of the parameter *Result* indicates the reason for failure with an appropriate error code.

6.4.6 The function GPCI_Get

After establishment of a local link to the Provider, the User may issue the GPCI_Get function to receive information:

GPCI_Get (*LinkId, Timer, IDU, Result*)

The parameter LinkId (conditional):

The value of the parameter *LinkId* (input parameter) is the unique identification of the established link between User and Provider. The value of this parameter is the same as that of the parameter *LinkId* of the GPCI_Link function. If only one link can be established at a time, the use of this parameter is optional, otherwise it should be mandatory.

The parameter Timer (optional):

The value of the parameter *Timer* (input parameter) indicates how long the User wants to wait until it gets back control, if no information arrives in the meantime. The User specifies the time period to wait in seconds:

1) value = 0: no wait; the User gets back control immediately;

2) value > 0: number of seconds; the User gets back control after that number of seconds or if any information arrives in the meantime.

The use of this parameter is optional. Meaning:

- for the synchronous mode: if the parameter is not used, it is interpreted as an infinite wait until information arrives;

- for the asynchronous mode: the polling mechanism or the interrupt mechanism is used for synchronization of User and Provider. Therefore, if this parameter is not used, it is interpreted as if the parameter has the value 0; the User gets back control immediately.

The parameter IDU (mandatory):

The IDUs exchanged with the function GPCI_Get belong to one of the three planes. The value of the parameter *IDU* (output parameter) of the GPCI_Get function is a complex data structure. This parameter is composed by n *IDUItem*s (n > 0). The meaning and data structure of the parameter *IDUI* may be specified by the designer of a standardized PCI depending on:

- the scope of that PCI;
- the location of that PCI in the communications system;
- the service(s) offered by the Provider;
- the accessible physical network(s).

Therefore, as an example, *IDUItem*s building the IDU may be:

a) IDUId: unique identification of the IDU; predefined by the designer of a standardized PCI;

b) ConnectionId: the User may use a local link to the Provider to establish one or more than one network connection to remote communications partners. The value of the parameter *ConnectionId* (input parameter or output parameter) identifies without ambiguity the connection on which an IDU arrives. If the item *ConnectionId* is used, a PCI specification should define, for the phase of connection establishment to a peer communication partner, who generates and maintains the *ConnectionId*s: the User and/or the Provider. Special attention should be paid to the case when a remote communication partner establishes the network connection (incoming call).

A PCI specification, which is in line with this ETR, should define the meaning and the data structure of the *IDU*s and the data type of each *IDUItem*. The ASN.1 description (see ITU-T Recommendation X.208 [3]) or any other FDT may be used.

The parameter Result (mandatory):

The value of the parameter *Result* indicates if the GPCI_Get function was accepted. On successful completion, the information should be conveyed to the User. Otherwise the value of the parameter *Result* indicates the reason for failure with an appropriate error code.

6.4.7 The function GPCI_Poll

Using the polling mechanism for synchronization of User and Provider, the GPCI_Poll function may be employed. This function can be used in two different ways:

1) the User calls the function and gets back control immediately and goes on working, even if no information is available;

2) the User calls the function and gets back control immediately, if information is available, otherwise the function is pending until information arrives; the maximum time period to wait for information is specified by the User.

GPCI_Poll (*LinkId, Timer, IDUList, Result*)

The parameter LinkId (conditional):

The value of the parameter *LinkId* (input parameter) is the unique identification of the established link between User and Provider. The value of this parameter is the same as that of the parameter *LinkId* of the GPCI_Link function. If only one link can be established at a time, the use of this parameter is optional, otherwise it should be mandatory.

The parameter Timer (optional):

The value of the parameter *Timer* (input parameter) indicates how long the User wants to wait until he gets back control, if no information arrives in the meantime. The User specifies the time period to wait in seconds:

1) value = 0: no wait; the User gets back control immediately;
2) value > 0: number of seconds; the User gets back control after that number of seconds or if any information arrives in the meantime.

If the parameter *Timer* is not used, it is interpreted as if the parameter would have the value 0; the User gets back control immediately.

The parameter IDUList (mandatory):

There may be one, more or no information available for the polling User. The parameter *IDUList* (output parameter) is a complex data structure. As a result of the function GPCI_Poll, the *IDUList* should be conveyed to the User containing the:

- IDULength (total length of the arrived IDU) for each available information.

Other items are possible, e.g.:

- ConnectionId (unique identification of the network connection on which the information arrived).

A PCI specification in line with this ETR, should define the meaning and the data structure of the parameter *IDUList* and the data type of each item. The ASN.1 description (see ITU-T Recommendation X.208 [3]) or any other FDT may be used.

The parameter Result (mandatory):

The value of the parameter *Result* indicates if the GPCI_Poll function was accepted. On successful completion a list of one, more than one or no arrived information (see parameter *IDUList*) should be conveyed to the User. Otherwise the value of the parameter *Result* indicates the reason for failure with an appropriate error code.

GPCI_Poll and GPCI_SetSignal are exclusive functions. Therefore, the functions GPCI_Poll and GPCI_SetSignal should not be used for the same local link.

A standardized PCI specification should define, if the PCI User is able to select the sequence, in which it wants to receive the available information or if it gets the information in a sequence determined by the Provider (e.g. first-in-first-out).

6.4.8 The function GPCI_SetSignal

Using the interrupt mechanism for the synchronization of User and Provider, the User should specify a call back routine (interrupt routine) with the GPCI_SetSignal function. This enables the User to go on working without polling for arrived information (see subclause 6.4.7). The call back routine should be activated automatically each time information arrives.

GPCI_SetSignal (LinkId, EntryPoint, Result)

NOTE: The GPCI_SetSignal function is used only once by the User just after invoking GPCI_Link function.

The parameter LinkId (conditional):

The value of the parameter *LinkId* (input parameter) is the unique identification of an established link between User and Provider, to which the User assigns a call back routine. The value of this parameter is the same as that of the parameter *LinkId* of the GPCI_Link function. If only one link can be established at a time, the use of this parameter is optional, otherwise it should be mandatory.

The parameter EntryPoint (mandatory):

The call back routine is addressed with the value of the parameter *EntryPoint* (input parameter).

A PCI specification in line with this ETR, should define how the call back routine can be realized in different operating environments.

If the User is responsible for providing enough memory resources (see subclause 6.3) for the interchange of information, the call back routine should be invoked with:

- IDULength: total length of the arrived information.

Other information conveyed to the User any time the call back routine is activated may be possible, e.g.:

- ConnectionId: unique identification of the connection on which the information arrived.

A PCI specification in line with this ETR, should define the meaning and the data structure of the information conveyed to the interrupt routine.

The parameter Result (mandatory:)

The value of the parameter *Result* indicates if the GPCI_SetSignal function was accepted. On successful completion the call back routine is accepted. Otherwise the value of the parameter *Result* indicates the reason for failure with an appropriate error code.

GPCI_SetSignal and GPCI_Poll are exclusive functions. Therefore, the functions GPCI_SetSignal and GPCI_Poll should not be used for the same local link.

6.4.9 Error handling

Each function of the GPCI of this ETR is atomic. This means that it either carries out its assigned task in full and reports success, or fails to carry out the assigned task and reports failure. An appropriate value of the parameter *Result* for each issued function indicates if the function was completed successfully. Completion of a function means acceptance of the function at the interface:

1) invocation of the function is valid at the current state of the communication between User and Provider;

2) the information, presented as parameter list of the function, is constructed correctly;

3) the Provider is not in an exceptional or erroneous state.

The parameter *Result* is therefore used to realize a flow control mechanism at the interface.

Completion of a function does not mean, that the task requested by the User with a particular function is successfully performed by the Provider (e.g. send an IDU to a remote communication partner). This ETR provides a second flow control mechanism **after** the acceptance of the function: exceptional or erroneous situations inside the Provider (network congestion, no more memory resources available, invalid parameter value, etc.) which are announced with the Administration Plane IDU (see subclause 6.5.3).

If a function leads to more than one reason for failure, the one which may be reported is in the Provider's domain.

A PCI specification according to this ETR, should define the structure of the error codes, the meaning of each error code and the data type of the error codes. The ASN.1 description (see ITU-T Recommendation X.208 [3]) or any other FDT could be used to specify the parameter *Result*.

EXAMPLE: In this example the error codes are specified with the ASN.1 description (see ITU-T Recommendation X.208 [3]):

```
ErrorCode::= CHOICE  {
      basicErrorCode              [1] INTEGER  {
                                  success (0),
                                  mandatory-parameter-missing (1),
                                  invalid-parameter-combination (3),
                                  provider-not-available (4),
                                  unknown-Provider (5),
                                  user-error (6),
                                  authorization-failure (7),
                                  system-error (8),
                                  no-specific-error (9)  },
      PCISpecificCodes     [2] INTEGER (10 ... 1999),
                           -- to be defined by the designer of a standardized PCI
                           -- specification, which is in line with this ETR
      productSpecificCodes        [2] INTEGER (2000 ... 3999) }
                           -- to be defined by the supplier of a developed product,
                           -- which conforms to the standardized PCI specification
```

ASN.1 is used as a FDT. It is not to be seen as encoding rules for the error codes.

6.5 The Interface Data Units (IDUs)

The IDUs interchanged between User and Provider depend on the PCI specified in line with the GPCI:

- the scope of that PCI;
- the location of that PCI in a communications system;
- the offered service(s);
- the characteristics and capabilities supported by that PCI;
- the accessible physical network(s);
- the possibility to enable or disable network control and signalling;
- the supported planes.

An IDU transferred as parameter of the function GPCI_Put and GPCI_Get belongs to one of the three planes:

a) User Plane: peer-to-peer communication;

b) Control Plane: network control and signalling (only outband);

c) Administration Plane: administration and management aspects.

The User Plane and the Administration Plane are mandatory. The Control Plane is only used if the Provider enables access to outband signalling network(s). The Control Plane is, therefore, optional.

The definition of the IDU Ids uniquely and without ambiguity is the responsibility of the body specifying a PCI standard.

6.5.1 IDUs of the user plane

The IDUs of the User Plane are information related to the peer-to-peer communication by means of the telecommunications service(s) offered by the Provider. Therefore, the set of IDUs of the User Plane depends on the PCI which is specified in line with the GPCI:

- the scope of that PCI;
- the location of that PCI in a communications system;
- the offered telecommunications services;
- the characteristics and capabilities supported by that PCI;
- the accessible physical network(s);
- the possibility to enable or disable network control and signalling;
- the supported planes.

A PCI specification should provide a complete list of User Plane IDUs.

As an example related to the OSI model, the User Plane IDUs may be categorized by the service primitives at the upper layer service boundary in Request IDU, Confirmation IDU, Indication IDU and Response IDU.

A standardized PCI which is specified according to the GPCI should provide a complete specification of User Plane IDUs.

6.5.2 IDUs of the Control Plane

The IDUs of the Control Plane are the signalling and control events related to outbound communications networks. The Control Plane IDUs handle services as provided by the related (outband) network. Therefore, the set of IDUs of the Control Plane depends on the PCI which is specified in line with the GPCI:

- the scope of that PCI;
- the characteristics and capabilities supported by that PCI;
- the accessible physical network(s);
- the possibility, to enable or disable network control and signalling;
- the supported planes.

If the User (or a component of the User) has access to network control and signalling, it may establish the physical network connection with the Control Plane IDUs before the logical connection with the User Plane IDUs to a remote communication partner. During the lifetime of such a connection, the User should be able to handle all network control and signalling related IDUs as well as the User Plane and Administration Plane IDUs. A User should not release the (physical) network connection before it has released the logical connection to the remote communication partner.

A standardized PCI which is specified according to the GPCI should provide a complete specification of Control Plane IDUs.

6.5.3 IDUs of the administration plane

The IDUs of the Administration Plane refer to administration or management features of a standardized PCI. The Administration Plane IDUs are related to the management the Provider and to the coverage of all additional facilities supported at a PCI. Therefore, the set of Administration Plane IDUs depends on the PCI which is specified in line with the GPCI:

- the scope of that PCI;
- the offered services;
- the characteristics and capabilities supported by that PCI.

A standardized PCI which is specified according to the GPCI should provide a complete specification of Administration Plane IDUs.

Administration Plane IDUs cover special features, accessible to the User at a standardized PCI, which are not related to either the peer-to-peer communication or the network control and signalling aspects. However, one Administration Plane IDU should be mandatory for all PCI specifications according to this ETR, to report asynchronous, exceptional or erroneous situations in the Provider (flow control):

- ExceptionReport IDU: mandatory:
  The Provider should have the possibility to announce an exceptional or erroneous situation (cannot-receive-further-IDUs, communication-path-broken, no-more-memory-available, etc.), which may lead to a rejection of a previous accepted IDU.

The following list of IDUs of the Administration Plane are optional:

1) Trace IDU: This IDU enables the User to get status report information on the progression of issued function(s);

2) Redirection IDU: This IDU enables the User to redirect a received IDU from one component of the User to another component of the User;

3) ConfigureProvider IDU: This IDU enables the User to configure the components of the Provider (packet size, checkpoint size, functional units etc.);

4) Cancel IDU: This IDU enables the User to cancel the execution of a previous issued function;

5) Convert IDU: This IDU enables the User to convert data from one standardized alphabet X to another standardized alphabet Y;

6) Print IDU: Tthis IDU enables the User to print data on printer x;

7) AddressAccess IDU: This IDU enables the User to get the network address of a remote communication partner by addressing this partner with a logical name (phonebook access);

8) Additional IDU: Any other IDU as a real extension of the set of Administration Plane IDUs specified in this ETR.

7 Interface characteristics

A User calls upon Provider telecommunications services by using the GPCI Exchange Mechanism functions described in clause 6. Provider structure and complexity are hidden from the User. According to Provider needs, this clause provides some hints to PCI developers specifying the characteristics of a PCI. A standardized PCI should be in line with the GPCI framework. Each standardized PCI should follow the general rules provided here with respect to the scope and location of that PCI.

7.1 User and Provider description

Clause 5 defines the concepts of User, Provider and GPCI. This subclause develops the definition of the User and the Provider of a standardized PCI which is specified according to the GPCI.

7.1.1 Provider model

The Provider is an abstract entity in a system which offers services to the User. As defined in clause 5, the Provider may consist of one or more components. Each component is a telecommunications resource which provides one or more telecommunications service(s).

The following components are defined:

- Communication Applications;
- Provider Modules.

7.1.1.1 Communication Application (CA)

A CA offers and uses one or more telecommunications service(s). To be able to offer a telecommunications service conformant with a standardized PCI (upper PCI), a CA uses one or more telecommunications service(s) conforming to another standardized PCI (lower PCI).

Figure 2 illustrates the relationship among PCIs (upper and lower), telecommunications services and CAs.

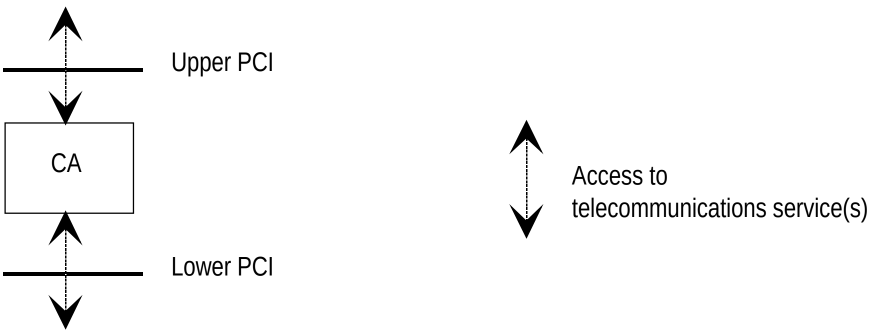


Figure 2: CA boundaries

7.1.1.2 Provider Module (PM)

A PM is an independent component which offers one or more telecommunications service(s) conformant with a standardized PCI (upper PCI); it relies on one or more physical medium(s).

Figure 3 illustrates the relationships between upper PCI, physical medium access, telecommunications services and PM.

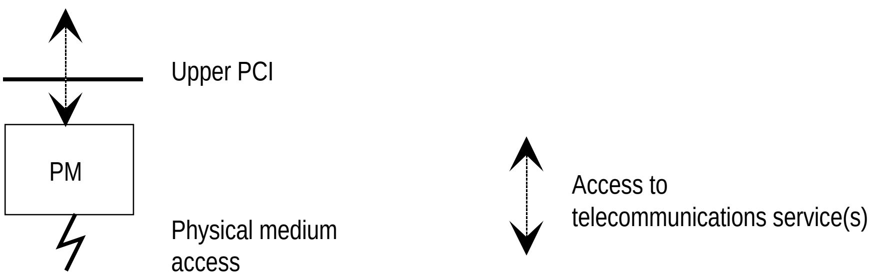


Figure 3: PM boundaries

7.1.1.3 Provider structure

A Provider may be:

- simple: built by a unique PM;
- complex: built by a set of CA(s) and PM(s) interacting through different standardized PCIs.

Figure 4 illustrates an example of a complex Provider.

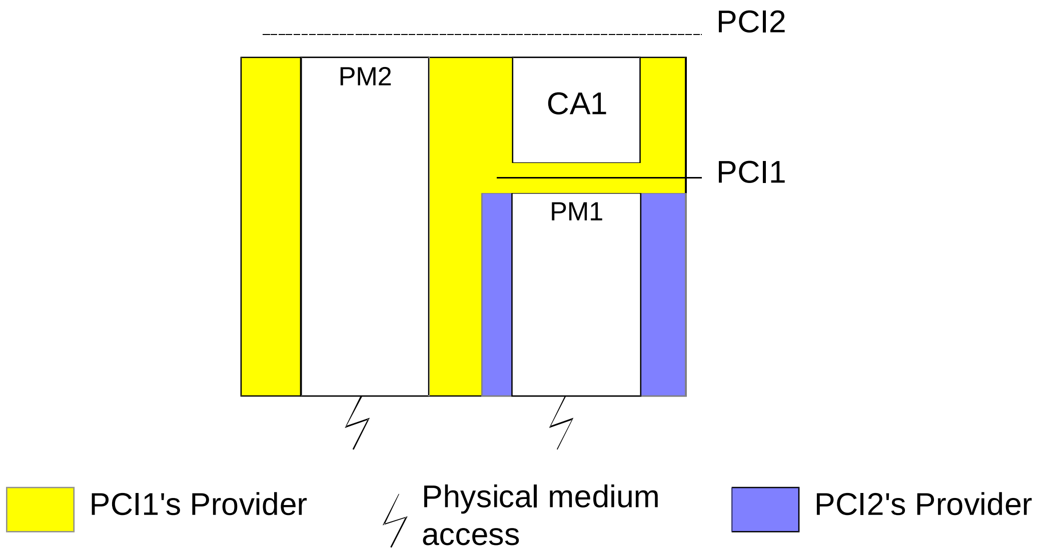


Figure 4: Provider example

In figure 4 PM2 and CA1 conform to the same standardized PCI (PCI2). They provide the same telecommunications services.

A PCI should allow redundancy as well as diversity of resources in the Provider. Thus, the Provider may contain a number of telecommunications resources offering:

- the same telecommunications service(s),
- different telecommunications service(s).

7.1.2 User model

The User is an abstract entity in a system which exploits the service of the Provider. As defined in clause 5, the User may be constituted of one or more components. The following types of components are defined:

- Local Applications;
- Communication Applications.

Local Application (LA)

An LA makes use of one or more telecommunications service(s) and offers one or more "*non-communication"* service(s) to other applications of the system (i.e. access to a database). An LA uses the telecommunications service(s) of a Provider's component in conformance with a standardized PCI (lower PCI).

Figure 5 illustrates the relationship between lower PCI, other services, telecommunications services and LA.

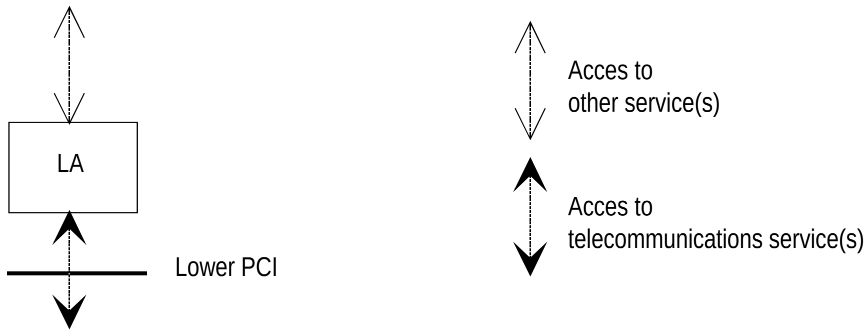


Figure 5: LA boundaries

Communication Application

As described in subclause 7.1.7, a CA is a Provider component which offers telecommunications service(s). A CA may also be a User component because it uses telecommunications service(s).

One User and one Provider interact through a unique standardized PCI. A CA conforms, at least, to two standardized PCIs (the lower and the upper PCIs of the CA). Thus, a CA is a component of its upper PCI Provider as well as being a component of its lower PCI User.

User structure

A User may be:

- simple: built by a unique LA;
- complex: built by a set of LAs and CAs interacting through different standardized PCIs.

Figure 6 shows an example of a complex User.

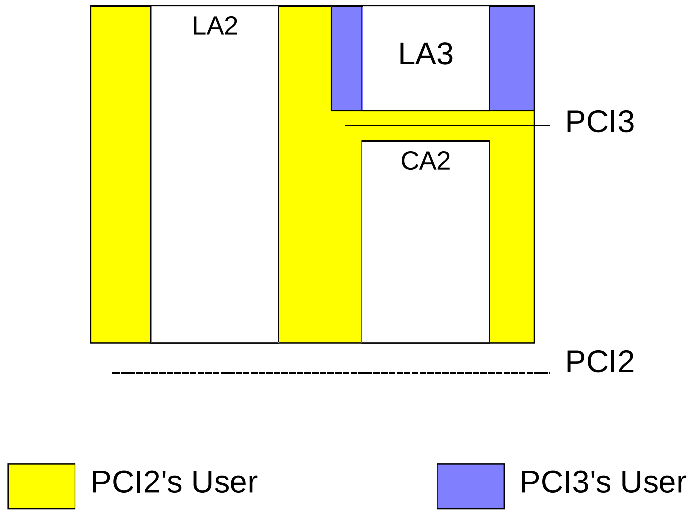


Figure 6: User example

In figure 6, LA2 and CA2 conform to the same standardized PCI (PCI2) and may use the same telecommunications service(s) offered by PCI2 Provider. CA2 is a component of PCI3 Provider as a component of the PCI2 User.

7.2 Data flow through the interface

The data flow through the interface represents the exchange of information between the User and the Provider. This information consists of the IDUs and the administrative parameters defined in clause 6.

PCIs which are specified according to the GPCI should allow interactions (LA/CA, LA/PM or CA/PM) between components realized by different manufacturers (interoperability as described in clause 5). In such cases, the data transfer between a User component and a Provider component should be realized as quickly as possible as they are part of the same application. Thus PCI specifications should describe an optimal way to exchange the data flow through the interface.

The data flow should not be processed, e.g. not interpreted, not reorganized, not modified as far as possible. The volume of the information should be reduced as far as possible to increase the data flow interchange rapidity. Thus PCI implementers should deeply define the IDU structures and contents, in accordance with clause 6.

Two main IDU management methods lead to interchange information through a PCI: the memory management and the file management.

7.2.1 Memory management

This subclause treats the case of interfaces using buffers of memory to interchange IDUs.

Depending on the architecture and infrastructure of the system, memory organization may be:

1) the User and the Provider share the same memory area. Thus the Provider may read and write in the memory of the User and "vice versa". In this case, the most efficient way to exchange IDUs is to avoid copying buffers;

2) the User and the Provider work in their own memory area. Each information message should be copied from the User memory area to the Provider memory area or from the Provider memory area to the User memory area. In this case the copy of the necessary information should be done as quickly as possible, e.g. using specific and optimized procedures.

7.2.2 File management

This subclause describes interfaces using files to interchange information. File management requires reading and writing data on disk(s). Performance decrease is feared.

Developers implementing this method should avoid writing duplicated information in different files or to keep unnecessary files on disk(s).

Such interfaces hinder efficiency within multi-host systems (copy of a file implies the transfer of this file from one host to another using Remote Procedure Call (RPC) procedures for example).

7.3 User/Provider relationship

A simple configuration of a telecommunications system, built in accordance with this ETR, allows one User component to interchange information with one telecommunications resource. This configuration is called a one-to-one relationship. A complex and powerful telecommunications system, also built by means of this ETR, may allow many User components to interact with many telecommunications resources. This configuration is called a many-to-many relationship.

This subclause specifies the characteristics of PCIs to allow one-to-one, many-to-one, one-to-many and many-to-many relationships between the User and the Provider.

A relationship between User and Provider implies one (or more than one) link.

7.3.1 One-to-one relationship

A PCI describing a one-to-one relationship allows only one User component to make use of one telecommunications resource at the same time.

Figure 7 illustrates the one-to-one relationship.

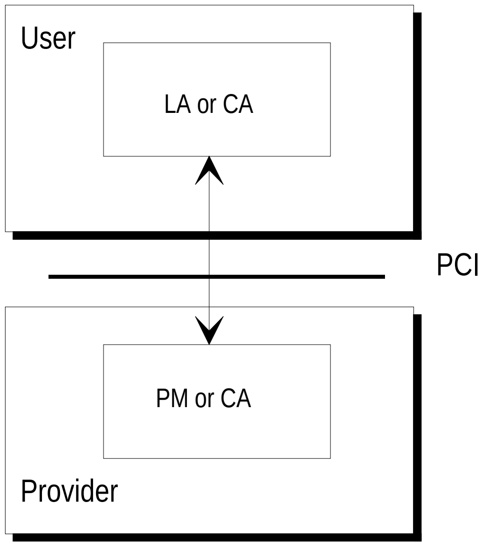


Figure 7: One-to-one relationship

The GPCI_ListResources function is optional for a one-to-one relationship PCI.

A one-to-one relationship PCI may progress to a more complex interface (upgrading), e.g. many-to-one, one-to-many or many-to-many relationship interface. Therefore, its specification should change in following the GPCI specification. If a system supports a one-to-one relationship the upgrade to a many-to-many relationship leads to a new specification, and an improvement of the one-to-one relationship. The configuration of the system changes by this upgrading process. Existing configuration(s) should be applicable as before.

7.3.2 Many-to-one relationship

A PCI describing a many-to-one relationship allows multiple User's components to make use of one telecommunications resource at the same time.

Figure 8 illustrates the many-to-one relationship.

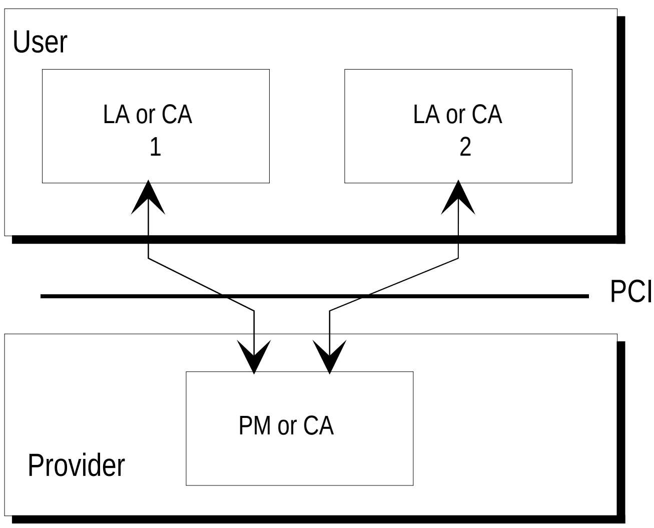


Figure 8: Many-to-one relationships

The number of relationships allowed by the Provider component is limited. Therefore, a many-to-one relationship PCI should specify how a Provider component refuses a new relationship. For example, a simple behaviour should be, at least, to return a congestion code to the GPCI_Link function.

The GPCI_ListResources function is optional for a many-to-one relationship PCI.

A many-to-one relationship PCI may progress to a more complex interface, e.g. many-to-many relationship interface. Therefore, its specification should progress according to the GPCI specification.

7.3.3 One-to-many relationship

A PCI which describes a one-to-many relationship allows one User component to use multiple telecommunications resources at the same time.

Figure 9 illustrates the one-to-many relationship.

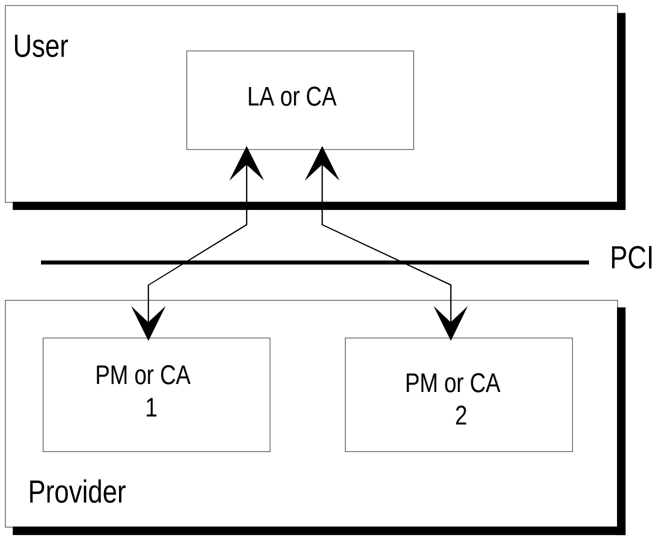


Figure 9: One-to-many relationships

The GPCI_ListResources function may be mandatory for a one-to-many relationship PCI.

A one-to-many relationship PCI may move to a more complex interface, e.g. many-to-many relationship interface. Therefore, its specification should evolve according to the GPCI specification.

7.3.4 Many-to-many relationship

A PCI describing a many-to-many relationship allows multiple Users components to use multiple telecommunications resources at the same time.

Figure 10 summarizes the possible relationships.

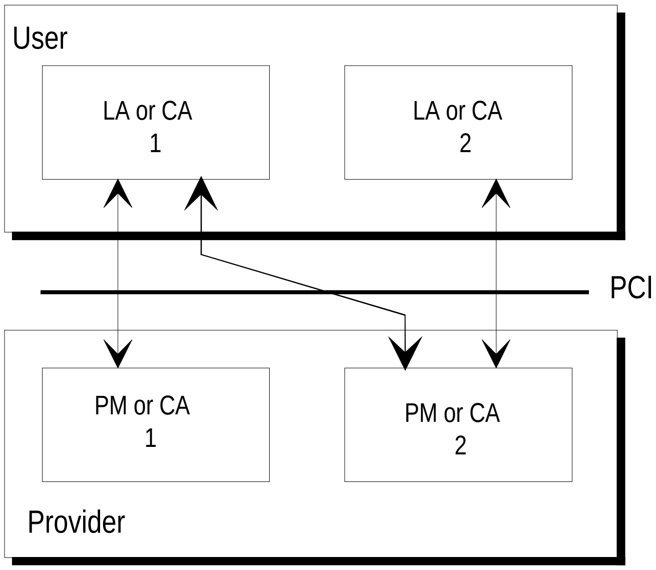


Figure 10: Many-to-many relationships

A standardized PCI, which is in line with this ETR and offers the many-to-many relationship, should describe the use of the parameter LinkId defined in the subclause 6.4.3.

A standardized PCI, allowing more than one PM and/or CA in the Provider, should specify how the different local links established between the User and the Provider are handled and processed.

The addition of telecommunications resources to the Provider should be hidden for the User. Thus a standardized PCI specification should clarify the evolution of the Provider when it grows from one telecommunications resource to many telecommunications resources.

7.3.5 Advise on GPCI_ListResources function

The function GPCI_ListResources offers the User the possibility to get a list containing the existing telecommunications resources of the Provider and their technical description (see subclause 6.4.2).

A standardized PCI specification should describe which element of the system realizes this feature and the technical means to achieve it.

The GPCI_ListResources function can be specified in different ways:

1) a dynamic way; the list of the existing telecommunications resources should be provided dynamically each time the function is called;

2) a static way; the list of the existing telecommunications resources could be read in a memory resource of the system. This solution requires that the common memory resource containing the list of the existing telecommunications resources should be updated each time a component is added or deleted in the system.

If the static way of the GPCI_ListResources function is selected, a standardized PCI specification, which is in line with this ETR, should define a mechanism to handle:

a) the addition of a new telecommunications resource in the Provider;

b) the deletion of an existing telecommunications resource from the Provider.

The addition in the Provider of a new telecommunications resource requires a procedure which updates the list of existing telecommunications resources. Updating the list means that a new ResourceRecord (see subclause 6.4.2) should be added to the list. The structure of this record, and the characteristics and capabilities of a telecommunications resource described in a record, should be specified by the designer of a PCI.

The ResourceRecord should be built of the following basic fields:

- the licence number of the resource;
- the identification of the resource;
- the version number of the resource;
- a reference indicating the telecommunications service(s) provided by this resource;
- a reference indicating the physical network available through this resource;
- quality of the service supported by this resource;
- restrictions (if any) to standardized services;
- any other important technical aspects of this resource.

Conversely, when a telecommunications resource is removed from the Provider, the record describing this resource should be deleted.

7.4 Co-ordination

Special attention should be paid to the physical network control and signalling aspects (outband only) used in a telecommunications system.

A standardized PCI, which is specified according to this ETR and provides access to network control and signalling, should clarify how the co-ordination between the physical and logical network connection is performed and who provides this co-ordination.

Network control and signalling aspects are related to the Control Plane. Either the User or the Provider should establish, control and release the physical network connection. Thus the co-ordination should be handled either by the User or the Provider.

The User should decide if the network control and signalling aspects are under its control. Therefore, the User selects the use of the Control Plane IDUs when it links to the Provider (see subclause 6.4.3).

The component, which realizes the co-ordination aspects, is called the co-ordination handler in the following subclauses.

7.4.1 Co-ordination handled by the User

In this case, the co-ordination handler is located in the User. The PCI should specify how the User handles Control Plane IDUs, User Plane IDUs and Administration Plane IDUs.

Depending on the structure and complexity of the Provider, the User may:

a) establish one local link with a telecommunications resource handling the User Plane, the Control Plane and the Administration Plane;

b) establish two local links, one with a telecommunications resource handling the User Plane and one with a telecommunications resource handling the Control Plane. The Administration Plane may be managed by one of these telecommunications resources or by both.

Figure 11 illustrates the two previous possibilities.

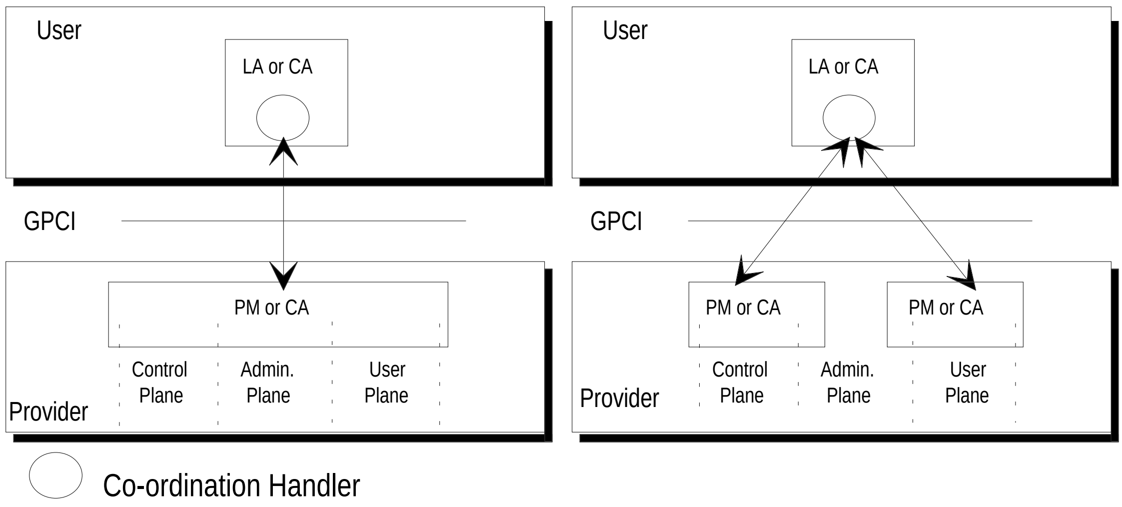
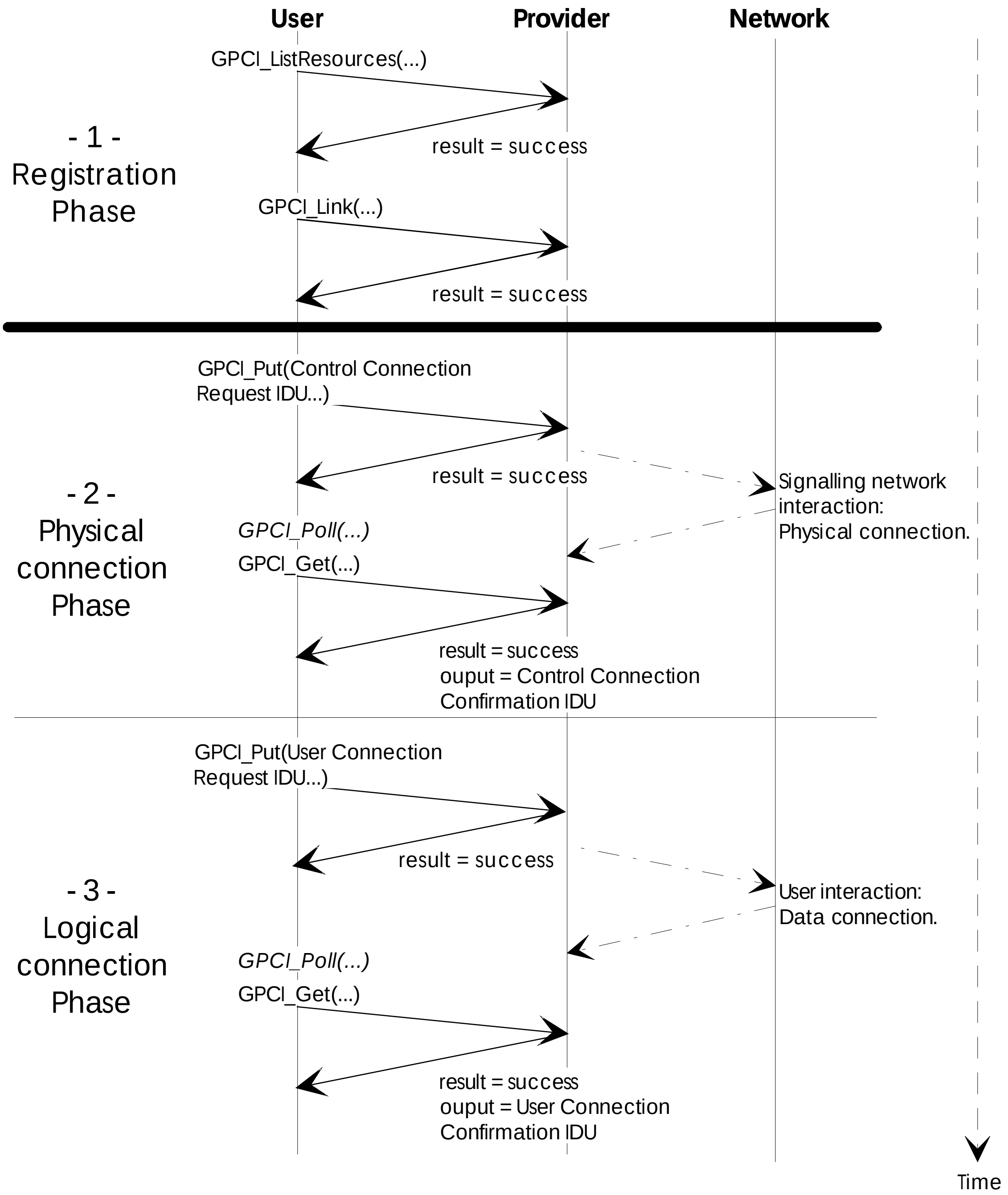


Figure 11: Co-ordination handled by the User

If the User handles the network control and signalling aspects, it may disconnect the peer-to-peer communication without releasing the physical network connection in order to use the established physical connection for another peer-to-peer communication.

Figure 12 illustrates the behaviour at a PCI if the User takes charge of the co-ordination. This example illustrates the valid behaviour for a communication based on a logical connection oriented protocol. The synchronization mode used is the polling mechanism.


User
Provider
Network
GPCI_ListResources(...)
result = success
- 1 -
Registration
Phase
GPCI_Link(...)
result = success
GPCI_Put(Control Connection
Request IDU...)
result = success
Signalling network
interaction:
Physical connection.
- 2 -
Physical
connection
Phase
GPCI_Poll(...)
GPCI_Get(...)
result = success
ouput = Control Connection
Confirmation IDU
GPCI_Put(User Connection
Request IDU...)
result = success
User interaction:
Data connection.
- 3 -
Logical
connection
Phase
GPCI_Poll(...)
GPCI_Get(...)
result = success
ouput = User Connection
Confirmation IDU
Time

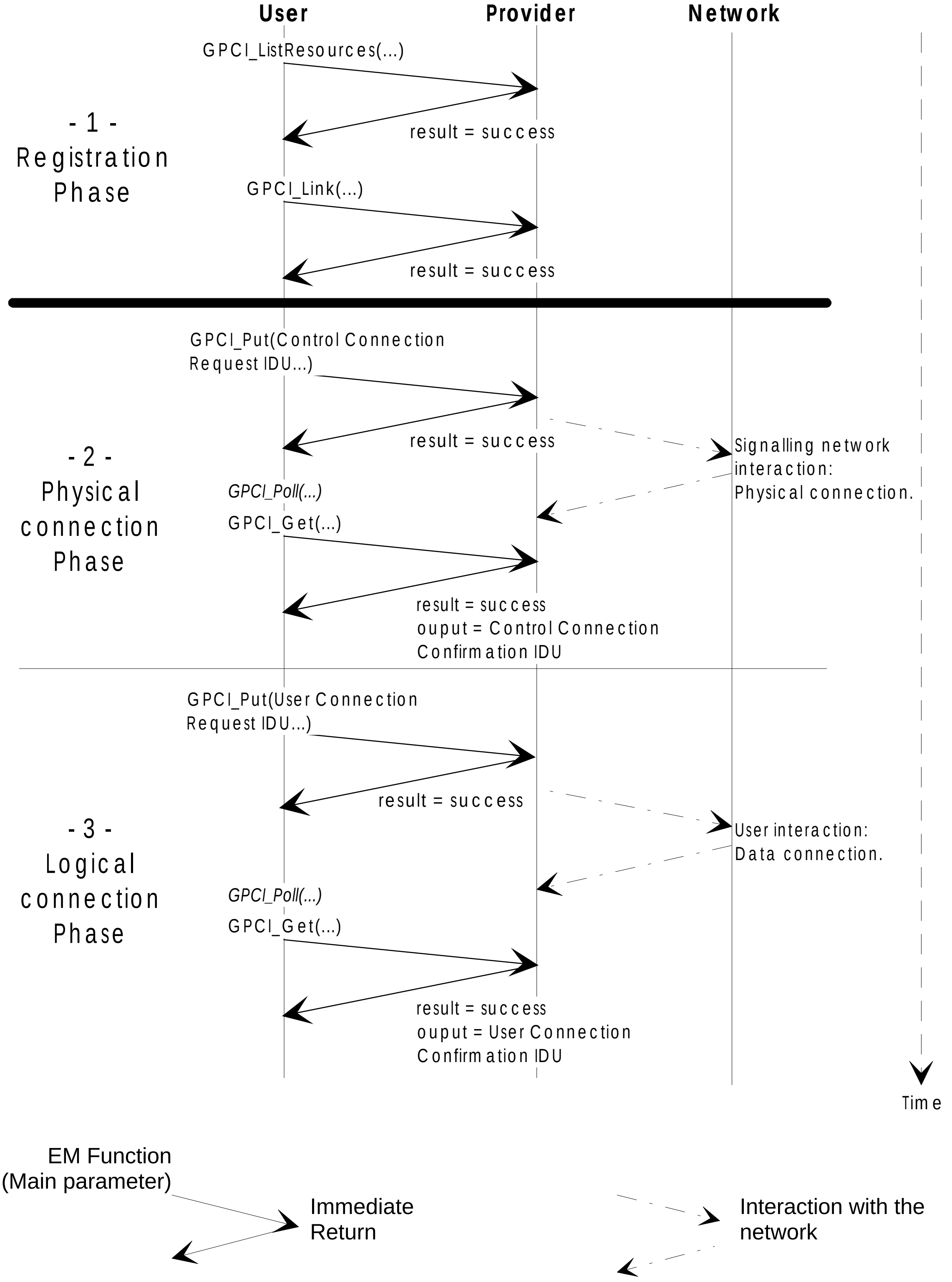


Figure 12: Example of a connection establishment sequence with co-ordination on the User side

7.4.2 Co-ordination handled by the Provider

In this case, the co-ordination handler is located in the Provider. The User should not handle the Control Plane IDUs.

Figure 13 illustrates the location of the co-ordination handler in this case.

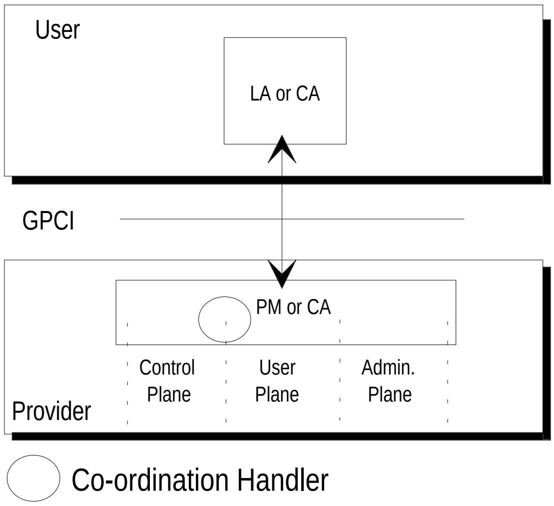


Figure 13: Co-ordination handled by the Provider

For such behaviour, PCI developers should specify how the User transmits to the Provider the global address of the connection, e.g. physical address of the remote host and the logical address of the peer application (see subclause 7.4.3).

The PCI should specify when the Provider should establish the physical connection. The time schedule may differ depending on the communication connection modes described in the following.

For a peer-to-peer communication based on connection oriented protocols, the following scenario takes place:

1) as the User does not use the Control Plane IDUs, the communication is established via a User plane IDU;

2) when the Provider obtains this IDU, first of all the physical network is connected then the logical protocol;

3) **the two connections are confirmed**, the Provider answers the User with the connection confirmation IDU of the User Plane,
or
one of the connections is not confirmed, the Provider answers the User with the appropriate IDU of the User Plane, e.g. a Disconnect Indication IDU or an Exception Indication IDU or another IDU, depending on the protocol.

For a peer-to-peer communication based on connectionless protocols, the information exchanged through a PCI is a set of standalone IDUs. Standalone in the sense that each one contains the peer address of its destination. PCI developers of such cases may define when and how the Provider takes charge of the physical connection; this is to avoid duplicated connections.

If the Provider handles the network control and signalling aspects, it may release the physical network connection when the last peer-to-peer communication on this network connection is disconnected. However, a PCI specification which is in line with this ETR should clarify the behaviour in this case.

Figure 14 illustrates the behaviour at a PCI, if the co-ordination is taken under control of the Provider. This example illustrates the valid behaviour for a communication based on a logical connection protocol oriented. The synchronization mode used is the polling mechanism.

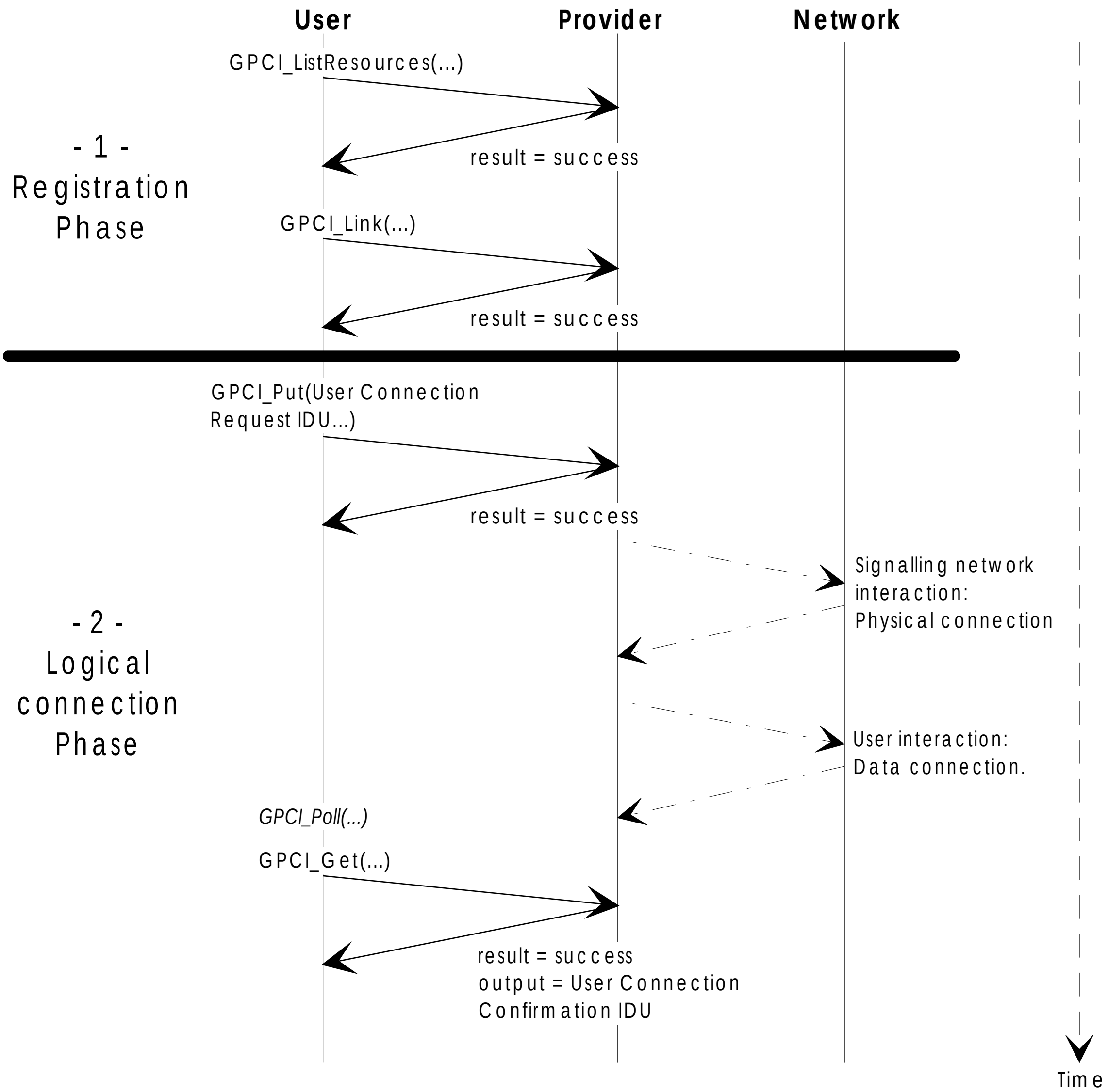


Figure 14: Connection establishment sequence with co-ordination on the Provider side

7.4.3 Network addressing

Wherever the co-ordination handler is located, the User should be responsible for providing the useful address(es) to establish the communication. PCIs which allow the Provider to handle the physical connection could specify how the Provider gets the values of the address(es).

Communication over outband signalling networks needs an initial connection of the physical layer and thus needs the remote equipment address (physical address). If this communication makes use of a protocol oriented connection, a logical connection is established using the peer application address (logical address).

To guide PCI developers, the following examples illustrate how the User may give the address(es) needed by the Provider to establish the communication:

EXAMPLE 1: The User creates Network Connection Objects (NCOs) containing all the necessary address information (physical, logical or both) to establish a connection. The User conveys the NCO to the Provider. The NCO is stored in both parts with the same reference. When a connection needs to be established the User gives the reference of an existing NCO to permit the Provider to establish the connection (physical, logical or both).

EXAMPLE 2: User and Provider share a common address book. Each record of this address book contains the necessary information to establish a connection (physical and logical). References to those records are accessed identically either by the User or the Provider. When a connection needs to be established the User provides the reference of an existing record to enable the Provider to establish the connection (physical, logical).

EXAMPLE 3: Physical and logical addresses are transferred through a User Plane IDU as an IDU item. If the PCI specifies a connection protocol oriented interface, this item may be part of a called address extension parameter in the IDU which initiates the logical connection, e.g. User Plane's Connection Request IDU.

7.5 Co-operation

The co-operation is an optional PCI feature which allows the User to dynamically incorporate a temporary telecommunications resource during an established connection to a remote communication partner, e.g. without releasing the connection.

This temporarily activated telecommunications resource should be able to interact with the Provider by means of the PCI standard, which is specified according to this ETR, whereas the communication between the User and this new telecommunications resource may follow different rules.

EXAMPLE: An LA is an interactive emulation of a remote host's terminal. A Provider Module offers an access to a Packet Switched Data Network (PSDN) network. Both (LA and PM) respect a PCI which is specified according to the GPCI. The LA uses the service of the PM to connect to a host, to log and to consult the menus of this remote system. If the LA needs to transfer a file from the host it activates a specific file transfer protocol. This protocol is handled by a new telecommunications resource which is a CA conforming to the same PCI. The CA is activated only during the file transfer, it takes on the role of the temporary telecommunications resource.

Figure 15 illustrates the temporary insertion of a telecommunications resource, i.e. co-operation, inside the peer-to-peer communication.

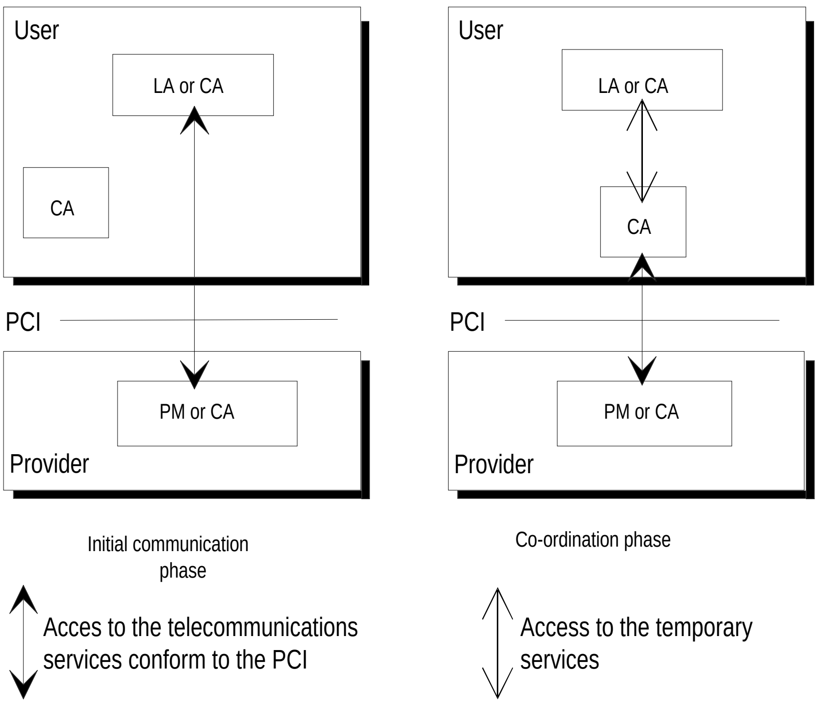


Figure 15: Example of co-operation during a connection

Co-operation of temporary telecommunications resources may be useful at any level of communication.

General rules to start and to stop a co-operation are provided in the following.

Starting mechanism

The co-operation is always initiated by the User (even if a remote request, for the integration of a temporary protocol in the communication, had been received).

At the start a User component (LA) establishes a local link with a telecommunications resource (PM) of the Provider and a peer-to-peer connection with a remote communication partner.

To start the co-operation phase the following steps could be performed:

1) the LA establishes a local link with a temporary telecommunications resource (CA);

2) the LA informs the CA that it should use the initial PM;

3) the CA links with the PM;

4) the CA informs the PM, that it wants to use the already established connection of the initial LA;

5) the PM accepts or rejects this request;

6) the CA informs the LA if the co-operation is accepted (depending on the result of point 5).

If the co-operation is agreed, the initial User may communicate with the temporary telecommunications resource by interchanging the appropriate User Plane IDUs.

During the co-operation, the LA should not use its local link to the PM.

Stop mechanism

The co-operation should be terminated by the telecommunications user when it no longer needs the CA. To stop the co-operation phase the following steps should be performed:

1) the LA requests the termination of the co-operation to the CA;

2) the CA informs the PM that it unlinks and that the LA maintains the peer-to-peer connection;

3) the PM accepts or rejects this request;

4) if the PM accepts the termination of the co-operation, the LA releases the local link to the CA.

The release of the link with the CA may be impossible because a transfer is still in progress.

7.6 Synchronization

Synchronization is an optional feature of standardized PCIs which allows a telecommunications resource to multiplex different channels of a physical medium.

EXAMPLE: A PM, which offers a basic Integrated Services Digital Network (ISDN) access, provides two connections through the B-channels and may provide one faster connection if it is synchronized with the two B-channels.

This feature is not a physical network capability. It should be provided by an entity of the system (certainly in the Provider) by the means of a specific protocol.

If a standardized PCI supports synchronization it should detail how to realize this mechanism.

7.7 Multiple interfaces

A telecommunications system may encompass multiple PCIs:

- hierarchical interfaces;
- parallel interfaces;
- and a combination of hierarchical and parallel interfaces.

7.7.1 Hierarchical interfaces

Hierarchical interfaces in a system are a set of PCIs located at different levels of the standardized telecommunications layers. Components which interact in an architecture with hierarchical PCIs create a telecommunications stack. This stack provides peer-to-peer communications.

A simple hierarchy involves two PCIs, one LA, one CA and one PM. Complex hierarchies involve three or more PCIs, one LA, two or more CAs and one PM.

Figure 16 illustrates a simple PCI hierarchy.

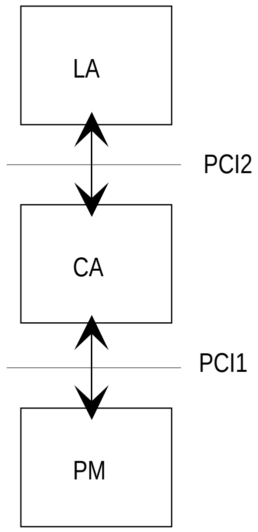


Figure 16: Hierarchical PCIs (simple example)

An LA should be located at the top of the stack. A PM should be located at the bottom of the stack and the CA(s) (one or more) are located between the LA and the PM.

7.7.1.1 Stack components configuration

As seen in clause 6, a standardized PCI allows the User to configure the Provider using Administration Plane IDUs. Thus a component of a stack configures the component(s) which offers it telecommunications service(s), e.g. in a simple hierarchy, the LA configures the CA and the CA configures the PM.

The components of a hierarchical architecture may require a coherent configuration, e.g. the configuration of each component is handled by the component located at the top of the stack. Thus the LA should be able to configure the CA(s) and the PM which build the telecommunications stack.

The configuration of the components may be performed by profiles. Profiles are built with a set of configuration parameters for each component of the pile. They are conveyed through PCIs from one component to another. When it gets the profile, one component may configure the lower component(s) which offers it telecommunications service(s).

7.7.1.2 Bypass facility

In a hierarchical architecture, the bypass facility is the ability for an application to activate a communication which can convey data, without the use of the PCI, directly to an external equipment (e.g. video screen, telephone, speaker, etc.).

EXAMPLE: A system composed of a computer containing an ISDN board. An external telephone handset is connected to this board. The computer runs one LA which allows a human operator to handle the ISDN board. Thus the LA interacts with the device driver (PM) of the ISDN board through a standardized PCI which is specified according to the GPCI. Using the LA, the operator may request a connection to a remote recipient. After the vocal connection the correspondents can converse through their handsets. At the end of the conversation, the operator sends an order to the LA to release the ISDN connection.

Such a facility is in the scope of the GPCI. A PCI specification which allows bypass should define:

- the mechanism to realize the bypass;
- the roles of the involved components.

7.7.2 Parallel interfaces

A communications system may consist of multiple parallel interfaces offering access to equivalent or different standardized services.

Figure 17 illustrates a simple PCI's parallelism.

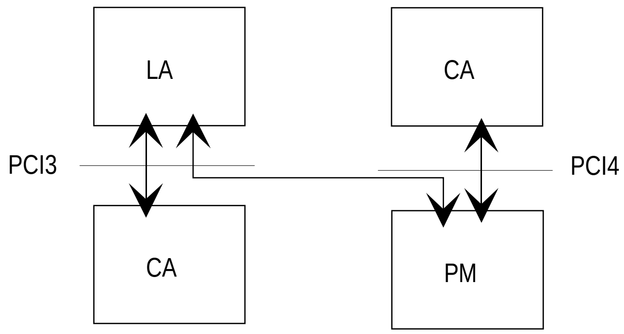


Figure 17: Parallel PCIs (simple example)

In a parallel architecture, PCIs may be activated at the same time without restrictions. An LA or a CA may use telecommunications services and a PM or a CA may offer telecommunications services as specified by standardized PCI(s).

7.7.3 Consequences of multiple interfaces system

The exchange mechanism is not necessarily the same for each PCI.

In a multiple PCI architecture, the User Plane and the Administration Plane should be mandatory but the Control Plane could be optional.

The Control Plane may be simplified to facilitate the access of the network control, especially for the connection and disconnection phases. A simplified subset of the Control Plane IDUs can be constructed.

The same Administration Plane may be used for all PCIs.

7.8 Flow control

The GPCI flow control is a method by which PCIs can regulate the exchange of information. It is independent of the possible flow control offered by the standardized service of this PCI.

Two GPCI functions allow data transfer between a User and a Provider:

- the GPCI_Put function to send information;
- the GPCI_Get function to receive information.

The User has three possibilities to know that received information are available to it (see clause 6):

- the use of the GPCI_Poll function with the synchronous mode;
- the use of the GPCI_Get function;
- the declaration of a call-back routine with the GPCI_SetSignal function.

7.8.1 Outgoing flow control

When the User sends information, an acknowledgement is received from the Provider. It indicates that the Provider has taken into account the request but successful processing is not guaranteed. If the Provider did not succeed in the handling of the information it should generate an ExceptionReport on this IDU. The User receives this event using the GPCI_Get function.

Figure 18 illustrates an error procedure in the outgoing flow information. The User handles the GPCI_Poll function mechanism to look for the available IDU(s).

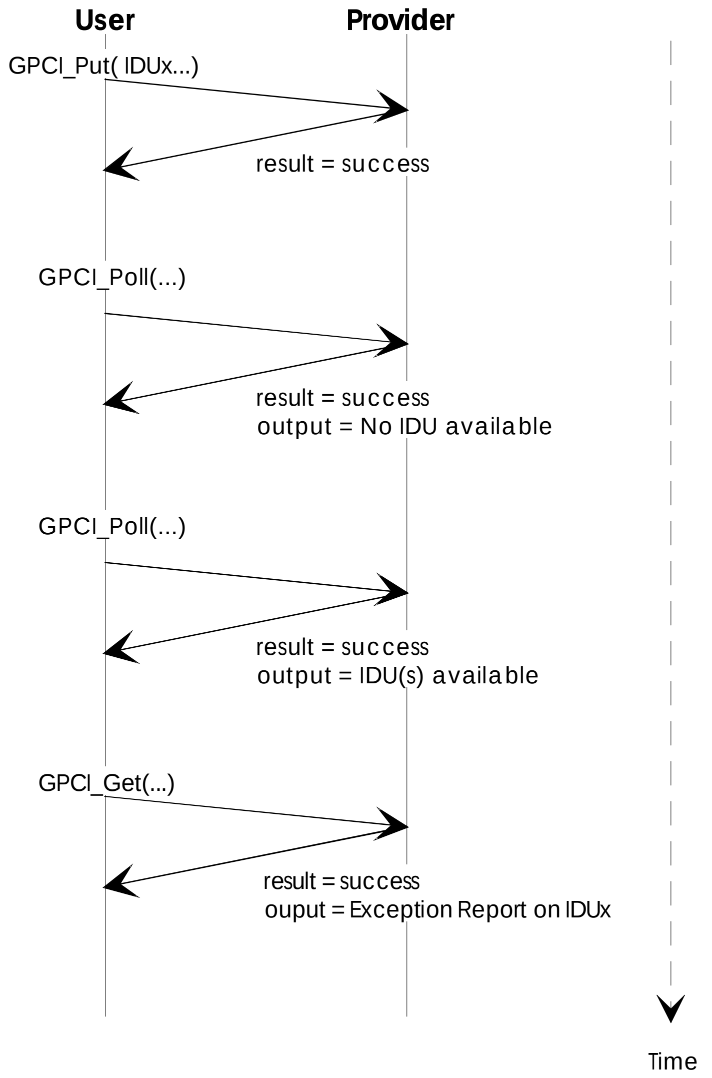


Figure 18: Anomalous data emission example

7.8.2 Incoming flow control

a) If the **polling mechanism is used with delay** its process is blocked until:

- an IDU becomes available then the User should use the GPCI_Get function to get it; or
- the delay is elapsed and the User should again use the GPCI_Poll function to check available IDU(s).

Figure 19 illustrates the incoming flow of information using the GPCI_Poll function with delay.

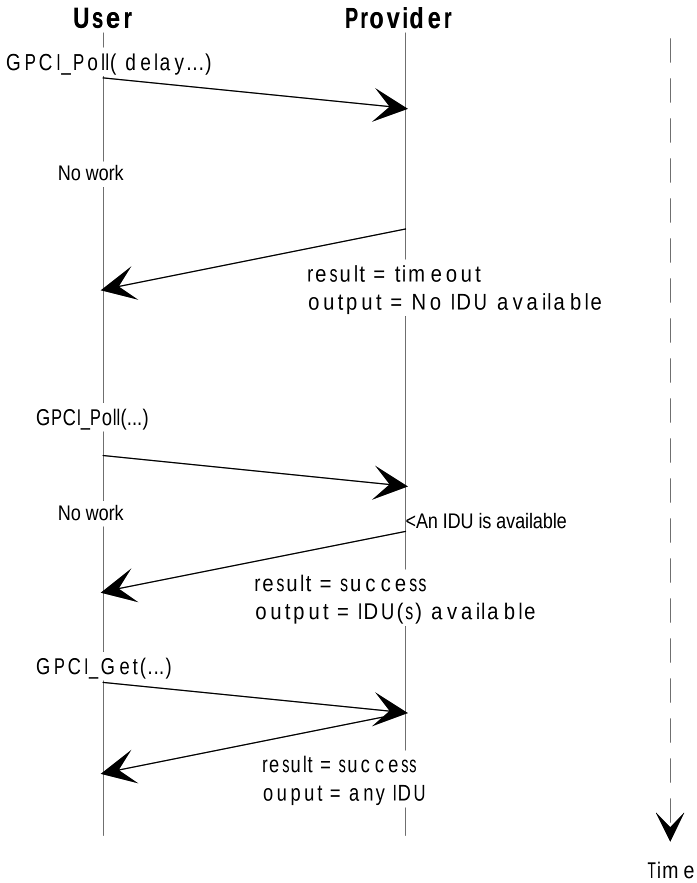


Figure 19: Data reception by polling with delay

b) If the User uses the **polling mechanism without delay** its process is not blocked. The GPCI_Poll result returns immediately and indicates:

- an IDU is available and the User should activate the GPCI_Get function to get it; or
- no IDU is available and the User should again use the GPCI_Poll function to examine the available IDU(s).

Figure 20 illustrates the incoming flow of information using the GPCI_Poll function without delay.

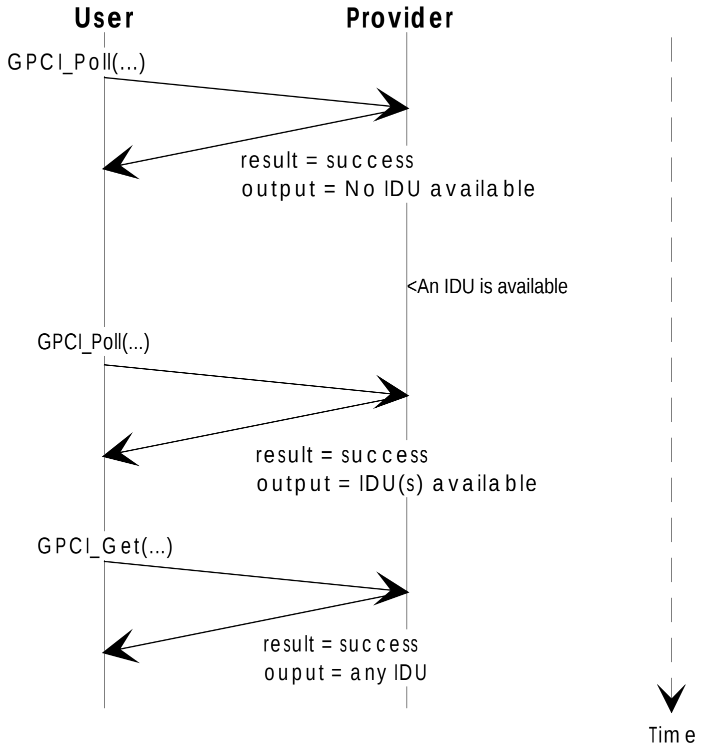


Figure 20: Data reception by polling without delay

c) If the User uses the **interrupt mechanism** the Provider may use a call-back routine to inform it when IDUs are available. If an IDU is available the User should use the GPCI_Get function to obtain it. The call-back routine should be as simple as possible and use the minimum of the system resources. Figure 21 illustrates the incoming flow of information using the call-back routine mode.

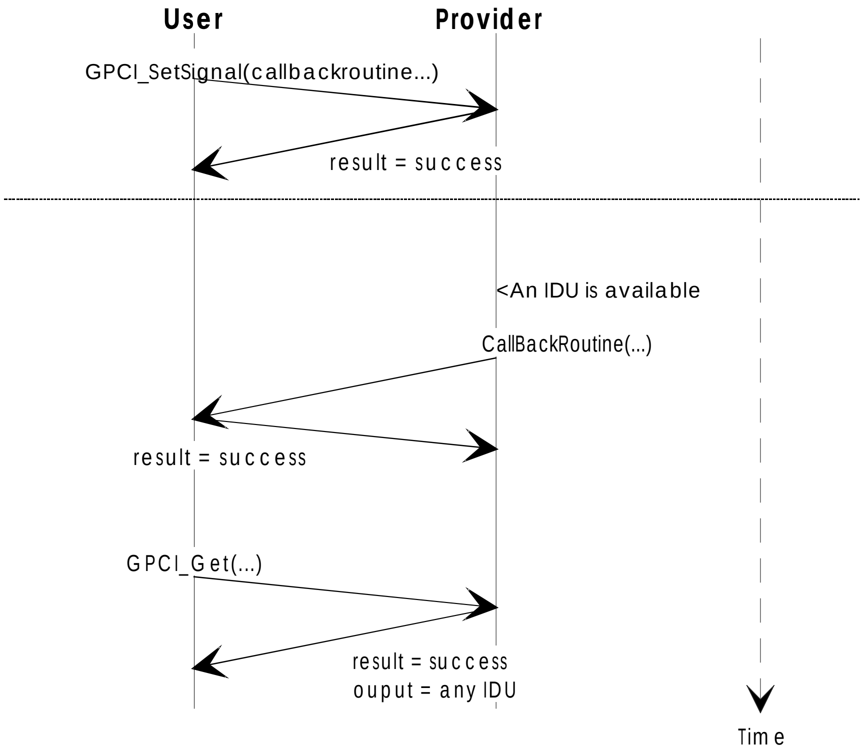


Figure 21: Data reception by interruption

The GPCI_SetSignal function is used only once by the User, just after invoking the GPCI_Link function.

d) If the User invokes the **synchronous mode** its process is blocked until:

- an IDU is available;
  or
- the delay is elapsed and the User should again use the GPCI_Get function to scrutinize available IDU(s).

7.8.3 Congestion

In a congestion situation, User or Provider may no longer be able to handle the information exchanged through the standardized PCI. The congested entity is blocked and IDUs may be lost.

A PCI in line with this ETR, should give general guidelines to solve these conflicts. The following examples may solve them:

EXAMPLE 1: Administration Plane IDUs are specified to stop the data flow just before congestion and to restart the data flow when the congestion is solved.

EXAMPLE 2: Administration Plane IDUs are specified to acknowledge each IDU passed through the PCI in both directions. Neither the User nor the Provider can send new information before the previous information has been acknowledged.

EXAMPLE 3: The Provider may stop the flow of information if it refuses to treat the GPCI_Put functions. Thus the result of this function indicates a failure.

EXAMPLE 4: The User may interrupt the conveyance of information if it stops using the GPCI_Get function.

8 PCI checklist

The model designed in this ETR is generic and identifies the relevant characteristics for a class of PCIs. The complete specification of a PCI, which is in line with the GPCI, should capture the interactions that can be performed at the interface, the observable states of the communicating objects (the components of a system) affected by the interactions and the circumstances in which the operations are appropriate. The PCI specification should be designed independently from any realization and should use a Formal Description Technique (FDT) to maximize clarity and minimize ambiguity, inconsistency and incompleteness.

A PCI specification should consist of:

a) the specification of the context within which the accessible (telecommunications) services operate;

b) the specification of the behaviour observable at the interface.

A PCI specification, modelled in line with the GPCI, should define:

1) the **scope** of the PCI: description of the need for and the purpose of this PCI;

2) the **field of application** of the PCI: description of the concerned telecommunications area, limitations and restrictions, if there are any;

3) the **infrastructure** supported by the PCI: description of the infrastructure of the communications system of this PCI (e.g. one computer or distributed computers; local and/or remote PCIs);

4) the **architecture** of the system: specification of the components of the system and their interrelationships;

5) **location** of the PCI: specification of the location of the PCI in the hierarchy of standardized layers in a communications system;

6) **services** of the PCI: specification of the provided telecommunications, administration and management services, as well as characterization of the accessible physical network(s); specification of restrictions in the use of the provided services, if there are any; the definition of profiles or classes for a PCI, if the are any.

A PCI specification should specify if special mode(s) of operation (transparent mode and/or bypass mode) are supported. It should be defined under which conditions those modes are applicable and all limitations and restrictions on the use of these modes, if there are any.

A PCI which is specified according to the GPCI should define the resource administration and management of the system. It should define:

- the used memory resources (file, buffer);

- proper or shared memory resources for User and Provider;

- which entity provides memory resources for the exchange of information and which entity releases this dynamically allocated memory resources;

- the limitations and restrictions concerning the telecommunications resources of the Provider, if any (e.g. a resource may be able to handle only a fixed number of local links);

- the limitations and restrictions concerning the physical network(s) accessible through the Provider, if any (e.g. a ITU-T Recommendation X.25 (see annex C) communication board may limit the number of physical connections to a peer communications partner at one time).

A PCI should give more details on operating system dependencies, programming language dependencies (language bindings), hardware platform dependencies. These details should be clearly distinguished from the PCI specification itself (e.g. in informative annexes or separate clauses).

A PCI should support all basic GPCI functions. If the Provider of a communications system offers only one telecommunications resource, the use of the function GPCI_ListResources is not meaningful. In this special case the PCI may not support the ListResources function. The designer of such a PCI should notice that the evolution of a "single resource Provider" to a "many resource Provider" is only possible by specifying a new PCI standard. This may lead to nonconformity between these standards.

A PCI may detect specific needs so that other functions, not specified in this ETR, become necessary. The definition of additional functions should be provided in a PCI specification if they are a real extension to the defined GPCI functions (no redundancy).

Each standardized PCI should define the data structures of the information, the data types of the parameters, the encoding rules and should specify the dynamic behaviour at that PCI with a formal specification language such as Specification and Description Language SDL (see ITU-T Recommendation Z.100 [2]) or any other FDT.

A standardized PCI specification should replace the preceding letters *GPCI_* of a function by a unique identification of that PCI.

8.1 The function GPCI_ListResources

As a result of the function GPCI_ListResources, a list of telecommunications resources is put at the User's disposal. A PCI which is in line with the GPCI should clearly define if the result of this function is the list of currently available or of existing resources of the Provider.

A PCI specification according to the GPCI should define the structure of a ResourceRecord, the meaning and the data type of each field of this record and all additional items, if there are any. Which field of a resource record is to be used to specify the query values of the parameter QueryMask should be defined.

A PCI, according to the GPCI, should specify the meaning and consequences of enabling network control and signalling (all signalling and control events are passed to the User, a part of them, etc.).

8.2 The function GPCI_Unlink

If the Provider cannot accept the GPCI_Unlink, he should have the possibility to reject the termination of the local communication. The User should be able to handle this rejection. In each PCI specification the situations and solutions of those conflicts should be defined.

8.3 The functions GPCI_Put and GPCI_Get

If the item ConnectionId is used in the parameter IDU a PCI specification should define, for the phase of connection establishment to a peer communications partner, who generates and maintains the ConnectionIds: the User and/or the Provider. Special attention should be paid to the case when a remote communications partner establishes the network connection (incoming call).

A PCI specification should define the meaning and the data structure of the IDUs and the data type of each IDUItem. ASN.1 (see ITU-T Recommendation X.208 [3]) or any other FDT may be used.

8.4 The functions GPCI_Poll

A standardized PCI specification should define whether the User is able to select the sequence in which he wants to receive the available information or if he gets the information in a sequence determined by the Provider (e.g. first-in-first-out).

8.5 The functions GPCI_SetSignal

A PCI specification should define the meaning and the data structure of the information conveyed to the interrupt routine.

A PCI specification, which is in line with this ETR, should define how the call back routine can be realized on different operating environments.

8.6 Error handling

A PCI specification, which is in line with this ETR, should define the structure of the error codes, the meaning of each error code, and the data type of the error codes. The ASN.1 description (see ITU-T Recommendation X.208 [3]) or any other FDT may be used to specify the parameter Result.

8.7 User plane IDUs

The IDUs of the User Plane are information related to the peer-to-peer communication by means of the telecommunications service(s) offered by the Provider. Therefore, the set of IDUs of the User Plane depends on the PCI which is specified according to the GPCI:

- the scope of that PCI;
- the location of that PCI in a communications system;
- the offered telecommunications services;
- the characteristics and capabilities supported by that PCI;
- the accessible physical network(s);
- the possibility to enable or disable network control and signalling;
- the supported planes.

The User Plane should be mandatory for each PCI specified in line with the GPCI. A standardized PCI should provide a complete specification of the User Plane IDUs.

8.8 Control Plane IDUs

The IDUs of the Control Plane are the signalling and control events related to outbound communication networks. The Control Plane IDUs handle services as provided by the outband network. Therefore, the set of IDUs of the Control Plane depends on the PCI which is specified in line with the GPCI:

- the scope of that PCI;
- the location of that PCI in a communications system;
- the offered telecommunications services;

- the characteristics and capabilities supported by that PCI;
- the accessible physical network(s);
- the possibility to enable or disable network control and signalling;
- the supported planes.

The Control Plane is optional for each PCI specified in line with the GPCI. If the Control Plane is supported, a standardized PCI should provide a complete specification of the Control Plane IDUs.

8.9 Administration Plane IDUs

The Administration Plane IDUs are used to manage the telecommunications service(s) and all additional facilities supported at a PCI. Therefore, the set of IDUs of the Administration Plane depends on the PCI which is specified in line with the GPCI:

- the scope of that PCI;
- the offered services;
- the characteristics and capabilities supported by that PCI.

The Administration Plane should be mandatory for each PCI specified in line with the GPCI. A standardized PCI should provide a complete specification of the Administration Plane IDUs.

8.10 PCI checklist proformas

The following proformas should be filled in by a PCI specification which is designed according to this ETR. The letters *<PCI_ID>* should be the unique identification of a PCI and should be defined in a PCI specification.

Table 1: Supported parameters of the <PCI_ID>ListResource function

<PCI_ID>ListResources parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
UserId					
QueryMask					
ResourceList	X				
Structure of Resource List: 1) NumberOfResources 2) ResourceId 3) ServiceId 4) NetworkId 5) NetworkControl 6) AdditionalItems:	X X X X X				
Result	X				

Table 2: Supported parameters of the <PCI_ID>Link function

<PCI_ID>Link parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
UserId					
ResourceId					
NetControlAccess					
LinkId					
Result	X				

Table 3: Supported parameters of the <PCI_ID>Unlink function

<PCI_ID>Unlink parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
UserId					
LinkId					
Result	X				

Table 4: Supported parameters of the <PCI_ID>Put function

<PCI_id>put parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
Linkid					
Idu	X				
Structure of the idu:					
Result	X				

Table 5: Supported parameters of the <PCI_ID>Get function

<PCI_ID>Get parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
LinkId					
Timer					
IDU	X				
Structure of the IDU:					
Result	X				

Table 6: Supported parameters of the <PCI_ID>Poll function

<PCI_ID>Poll parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
LinkId					
Timer					
IDUList	X				
Structure of the IDUList:					
Result	X				

Table 7: Supported parameters of the <PCI_ID>SETSIgnal function

<PCI_ID>SETSIgnal parameters	**Mandatory**	**Optional**	**Conditional**	**Not used**	**Data type**
LinkId					
EntryPoint	X				
Result	X				

Table 8: Supported parameters of additional <PCI_ID> function

<PCI_ID><function name> * parameters	Mandatory	Optional	Conditional	Not used	Data type
*: <function name> should be defined in a PCI specification according to this ETR					

Table 9: List of User Plane IDUs

IDU name	Mandatory	Optional	Conditional	Identification

Table 10: List of Administration Plane IDUs

IDU name	Mandatory	Optional	Conditional	Identification

Table 11: List of Control Plane IDUs

IDU name	Mandatory	Optional	Conditional	Identification

9 Testing aspects

Conformance relates an implementation (a product) to the standardized PCI. Any proposition that is requested in a PCI standard should be realized in an implementation developed in conformance to that PCI.

A PCI specification, which is in line with this ETR, should define a conformance testing methodology to enable the definition of test suites, which are either part of the PCI specification itself or described in a separate document. Test suites provide a number of test cases to support developers in the testing of their commercial products and to enable test laboratories to test different PCI products which claim conformance to a standardized PCI specification. A product, validated to be conformant, gives the confidence that the implementation has the required capabilities and that the communication at the PCI conforms consistently to the required behaviour.

Today the only international accepted conformance testing methodology, defining a common testing methodology, appropriate testing methods and procedures, is provided by ISO/IEC. The purpose of ISO/IEC 9646 [4] is to define the methodology, to provide a framework for specifying conformance test suites and to define the procedures to be followed during testing. In principle, the goal of conformance testing is to establish whether the implementation conforms to the relevant OSI standard. A PCI specification, of course, is not an OSI protocol, but the general aspects, the testing methods, and the purposes of conformance testing described in ISO/IEC 9646 [4] may apply also for standardized PCIs. However, adaptation of certain terms and notations to the specific requirements of PCIs are necessary.

As identified in this ETR, parts of a PCI specification are related to the individual operating environment (operating system, programming language, and hardware environment). Furthermore, the behaviour and conformance to a PCI standard cannot be observed outside the system because a PCI has only local impact. Therefore, the Points of Control and Observation (PCOs) are located within the System Under Test (SUT). In order to test a PCI comprehensively an upper tester and a lower tester may reside within the SUT as well. A test engine therefore needs operating environment specific test tools, which causes many more problems and costs than under circumstances where the behaviour can be observed from outside the SUT. These aspects should be taken into account when the test method and abstract test suites are defined for a standardized PCI.

9.1 Rules for PCI testing

Testing of a PCI means:

- static conformance testing: the features of a standardized PCI are supported in a product;

- dynamic behaviour check: the realization of information exchange and the results observable at that PCI;

- syntax check: the structure of the exchanged information.

PCI testing is product oriented. Therefore, conformance testing of a PCI, which is specified according to this ETR, means:

a) test of a realization/implementation of that PCI;

b) test of a realization/implementation of the PCI functions;

c) test of the structure and data types of the exchanged information;

d) test of the dynamic behaviour at the PCI (valid behaviour, invalid behaviour);

e) test of the access to the provided telecommunications service(s) across the PCI.

PCI conformance testing excludes:

- testing of the telecommunications service(s) of a Provider (i.e. protocol test);

- testing of the realization/implementation of a User interacting with a PCI product.

However, in certain circumstances, e.g. if the User itself is a product, which claims to be conformant to a standardized PCI, it may be meaningful to test a part of the realization/implementation of the User:

- to check that a User is able to invoke the functions provided at the PCI;
- to check that a User understands and acts properly to the result of these functions;
- to check that a User is able to build correctly the information exchanged with these functions.

The PCI conformance tests specified for a standardized PCI therefore depends on:

- the scope of a PCI;
- the location of that PCI;
- the services provided at that PCI.

A PCI standard, according to ISO/IEC 9646 [4], should specify:

1) the conformance requirements;
2) the PCI Implementation Conformance Statements (PCI ICS) proforma;
3) the PCI Implementation eXtra Information for Testing (PCI IXIT) proforma;
4) the appropriate Test Method;
5) the general rules for defining an Abstract Test Suite (ATS) with Tree and Tabular Combined Notation (TTCN);
6) probably the ATS(s) itself.

10 Notation and structure of a PCI

A PCI specification, which is in line with this ETR, should use a standardized FDT to define:

a) the dynamic behaviour at the PCI;
b) the data structures and data types used in the PCI specification.

This issue should:

- be easy to use and to interpret;
- allow the description of a PCI interface without ambiguity;
- be appropriated for telecommunications systems.

FDTs such as SDL seem to be appropriate to define the syntax and the semantics of a PCI specified in line with the GPCI. The syntax description should be used to specify the architecture of the communication system, i.e. components of the system (User, Provider, PCI, LA, CA, PM, the PCI functions, structure of the conveyed information etc.).

The semantic description should be used to define the behaviour (valid as well as invalid) of the components of a system. The sequence of the observable events in a normal behaviour and in defective behaviour should be defined.

The ASN.1 notation (see ITU-T Recommendation X.208 [3]) can be used for the data description.

Annex A: Clarification of the terms "API" and "PCI"

The abbreviation API stands for "Application Programmable Interface" which is a boundary between an application software entity (User) and an application platform (Provider). An API is directly, or indirectly, associated with a wide range of software interfaces which have in common the fact that they provide access to a computer system facility. These facilities may be related to communication services, but most of the APIs are concerned with the use and the control of local computer system resources (e.g. disks, memory, other special devices or system services).

The abbreviation PCI stands for "Programmable Communication Interface" and is associated with all those APIs which are able to manage OSI oriented or other communication services. A detailed classification is provided in the following clauses which makes it clear that the PCI has a widespread scope of application.

As PCIs are a subset of APIs, the general rules and guidelines for APIs, as given for instance in ISO/JTC1 document N2890 (see annex C), are also valid (when applicable) for PCIs.

A.1 Categories of APIs

The term API is used in a more general way, as mentioned above. The general subject may be divided into smaller topics as shown in figure A.1. The content of the different branches of that tree are described in the relevant subclauses.

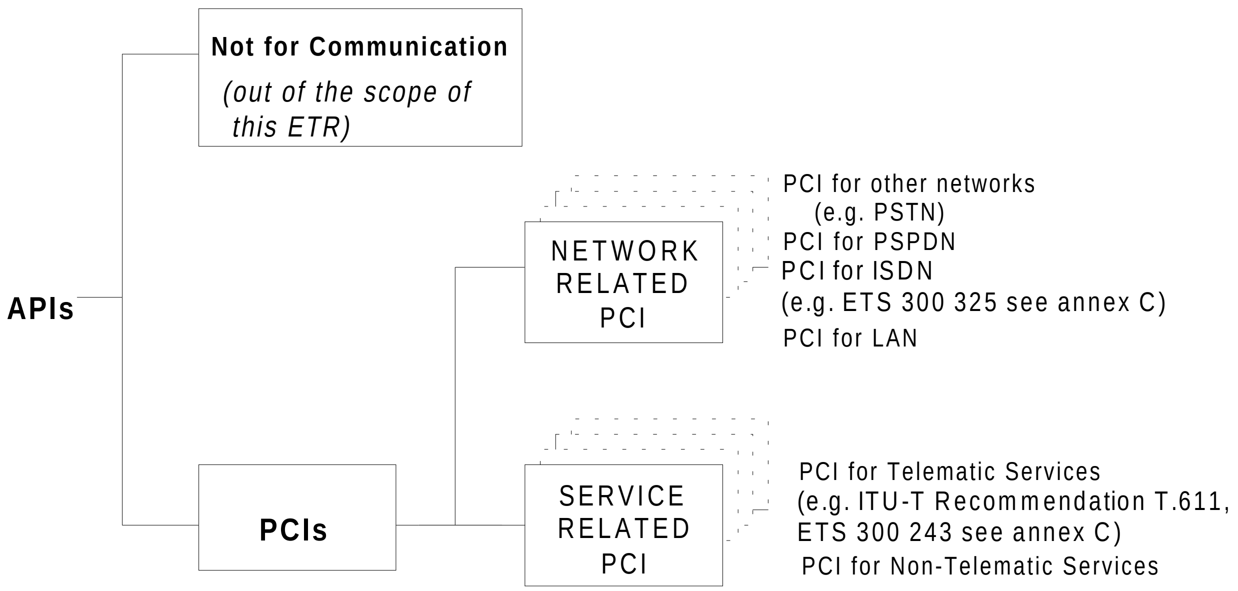


Figure A.1: Hierarchical structure of API categories

A.2 APIs for non-communication purposes

This category of API applies to service peripheral devices (e.g. printers, screens and other useful equipment), Programming language supports (software libraries, runtime environments) and user system services (X-Windows or Ms-Windows).

This category of API is very important and useful for the construction of data processing systems but is not necessary for data communications. Therefore, this group of APIs is outside of the scope of this ETR.

A.3 Programmable Communication Interfaces (PCIs)

This second category of interfaces covers telecommunications aspects.

The major requirements in order to define a PCI are:

a) several kinds of interfaces may be needed depending on the provided telecommunication services. The general functional model defines the location of the PCIs and how they may work together. Today two levels are recognized: *network related* and *service related* PCIs;

b) the PCIs need to be portable and thus, independent of hardware and operating systems;

c) the PCIs need to take into account all the ITU-T Recommendations relevant for the provided ITU service;

d) the PCIs need to allow extensions for facilitating the provision of private functions.

A.3.1 Network related PCIs

Network related PCIs contain programming interfaces which have been designed for the access of specific types of network. In the sense of OSI, those PCIs could be located at layer 3 (or below). Network related PCIs exist for various networks which may be public networks (e.g. ITU-T Recommendation X.25 (see annex C), ISDN, Public Switched Telephone Network (PSTN), etc.) but also private networks in case of Local Area Networks (LANs).

Network related PCIs facilitate the use of specific network capabilities and are thus strongly bound to the underlying network services.

A.3.2 Service related PCIs

This category of PCIs is concerned with the provision of application oriented services whereas the term "application" is related to layer 7 of the OSI model.

The services provided by applications related PCIs may be ITU oriented services such as Facsimile, Message Handling, File Transfer, Electronic Data Interchange (EDI), etc. Service related PCIs may also be identified in private solutions.

Annex B: Samples of functional behaviour at the GPCI: a SDL description

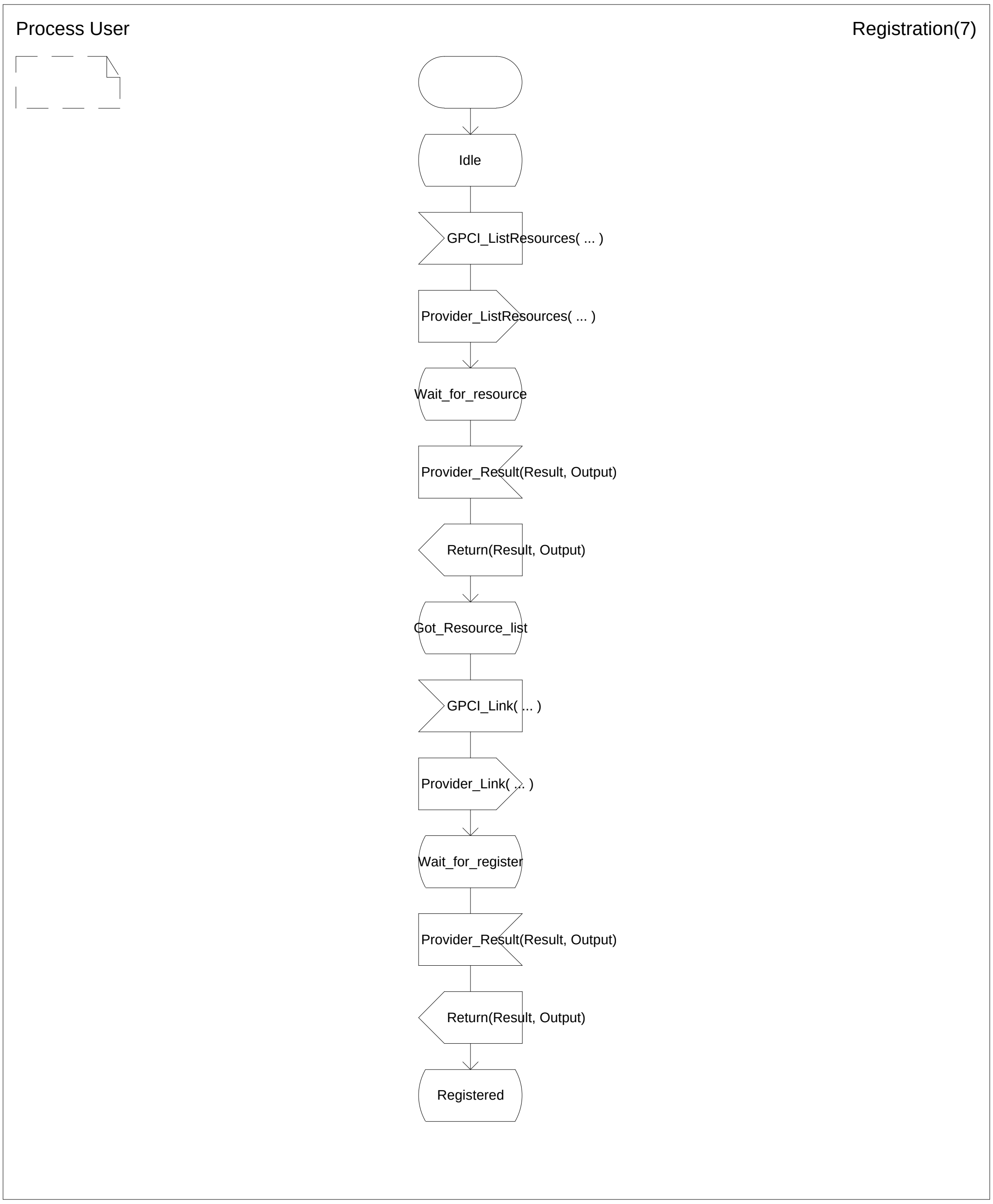


Figure B.1

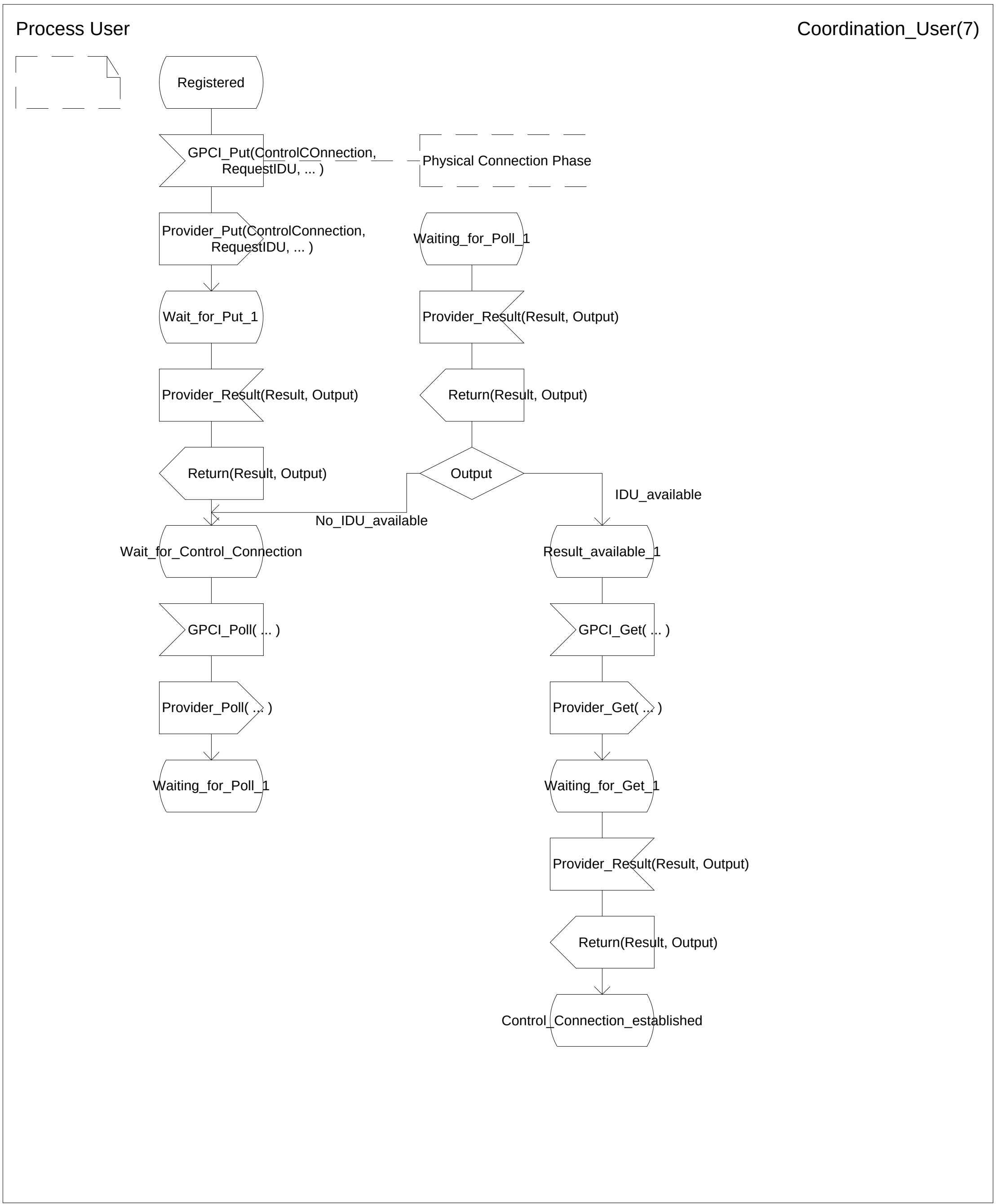

Process User
Coordination_User(7)
Registered
GPCI_Put(ControlCOnnection, RequestIDU, ...)
Physical Connection Phase
Provider_Put(ControlConnection, RequestIDU, ...)
Waiting_for_Poll_1
Wait_for_Put_1
Provider_Result(Result, Output)
Provider_Result(Result, Output)
Return(Result, Output)
Return(Result, Output)
Output
IDU_available
No_IDU_available
Wait_for_Control_Connection
Result_available_1
GPCI_Poll(...)
GPCI_Get(...)
Provider_Poll(...)
Provider_Get(...)
Waiting_for_Poll_1
Waiting_for_Get_1
Provider_Result(Result, Output)
Return(Result, Output)
Control_Connection_established

Figure B.2

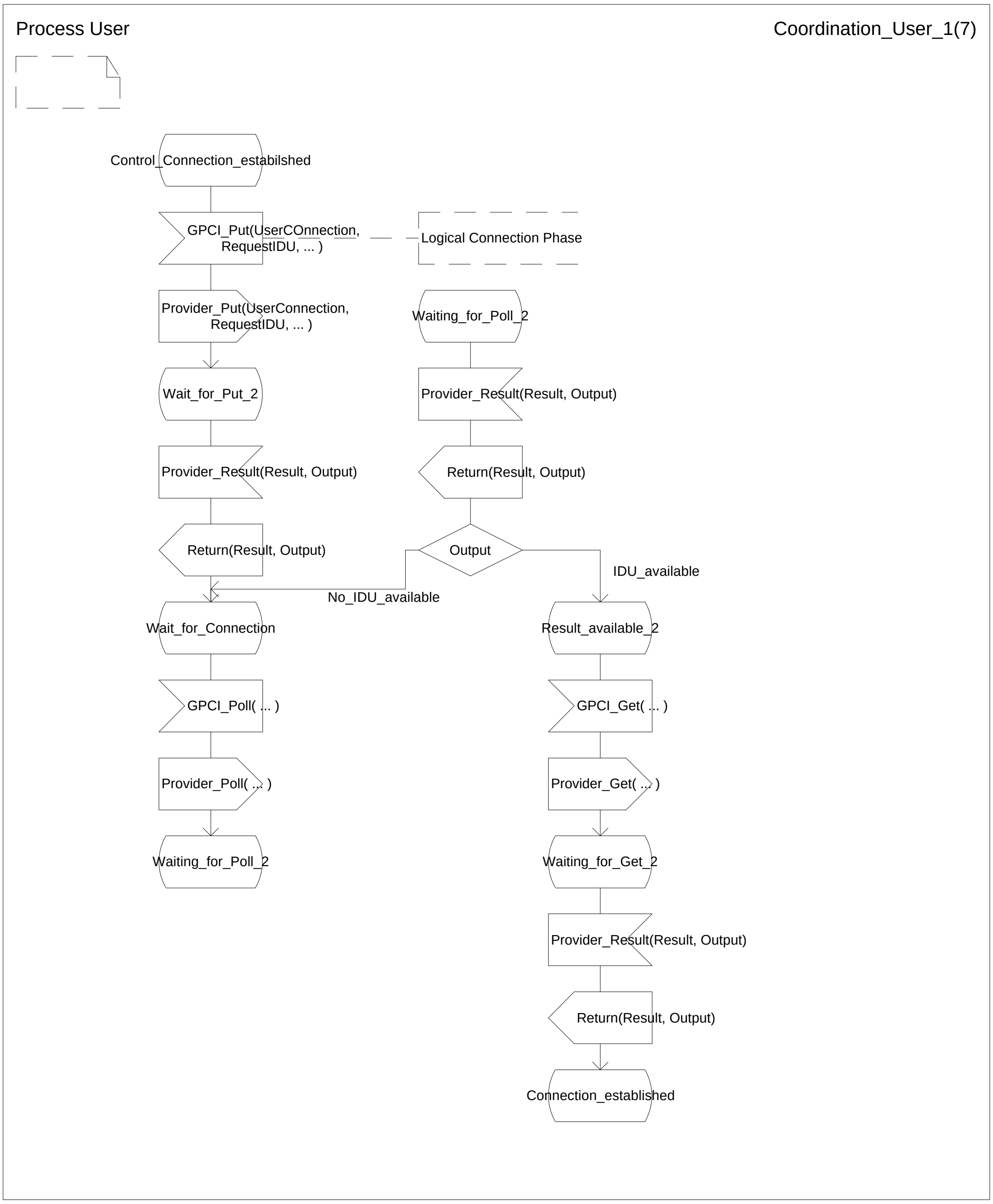

Process User
Coordination_User_1(7)
Control_Connection_estabilshed
GPCI_Put(UserCOnnection, RequestIDU, ...)
Logical Connection Phase
Provider_Put(UserConnection, RequestIDU, ...)
Waiting_for_Poll_2
Wait_for_Put_2
Provider_Result(Result, Output)
Provider_Result(Result, Output)
Return(Result, Output)
Return(Result, Output)
Output
IDU_available
No_IDU_available
Wait_for_Connection
Result_available_2
GPCI_Poll(...)
GPCI_Get(...)
Provider_Poll(...)
Provider_Get(...)
Waiting_for_Poll_2
Waiting_for_Get_2
Provider_Result(Result, Output)
Return(Result, Output)
Connection_established

Figure B.3

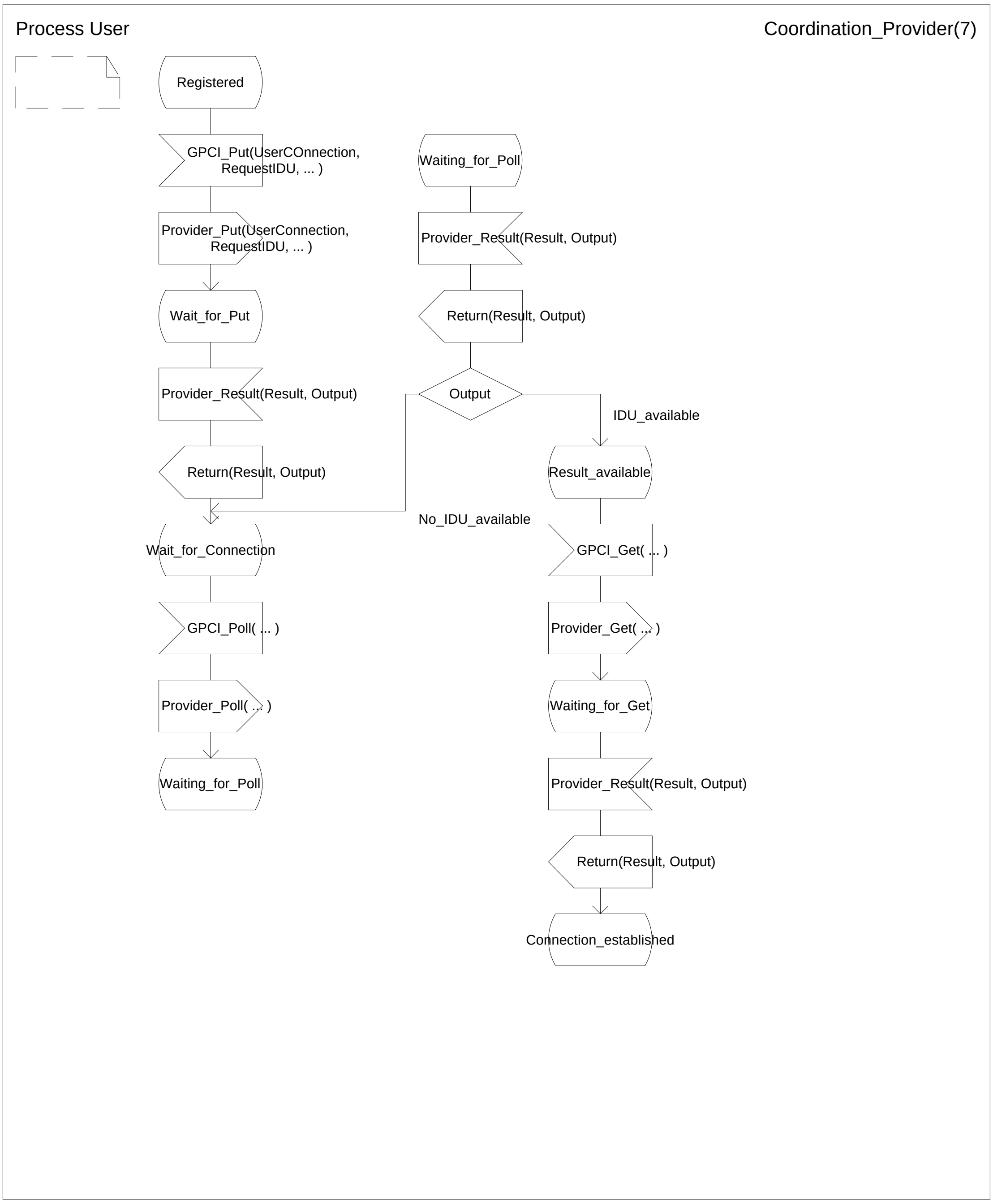

Process User
Coordination_Provider(7)
Registered
GPCI_Put(UserCOnnection, RequestIDU, ...)
Provider_Put(UserConnection, RequestIDU, ...)
Wait_for_Put
Provider_Result(Result, Output)
Return(Result, Output)
Wait_for_Connection
GPCI_Poll(...)
Provider_Poll(...)
Waiting_for_Poll
Waiting_for_Poll
Provider_Result(Result, Output)
Return(Result, Output)
Output
IDU_available
No_IDU_available
Result_available
GPCI_Get(...)
Provider_Get(...)
Waiting_for_Get
Provider_Result(Result, Output)
Return(Result, Output)
Connection_established

Figure B.4

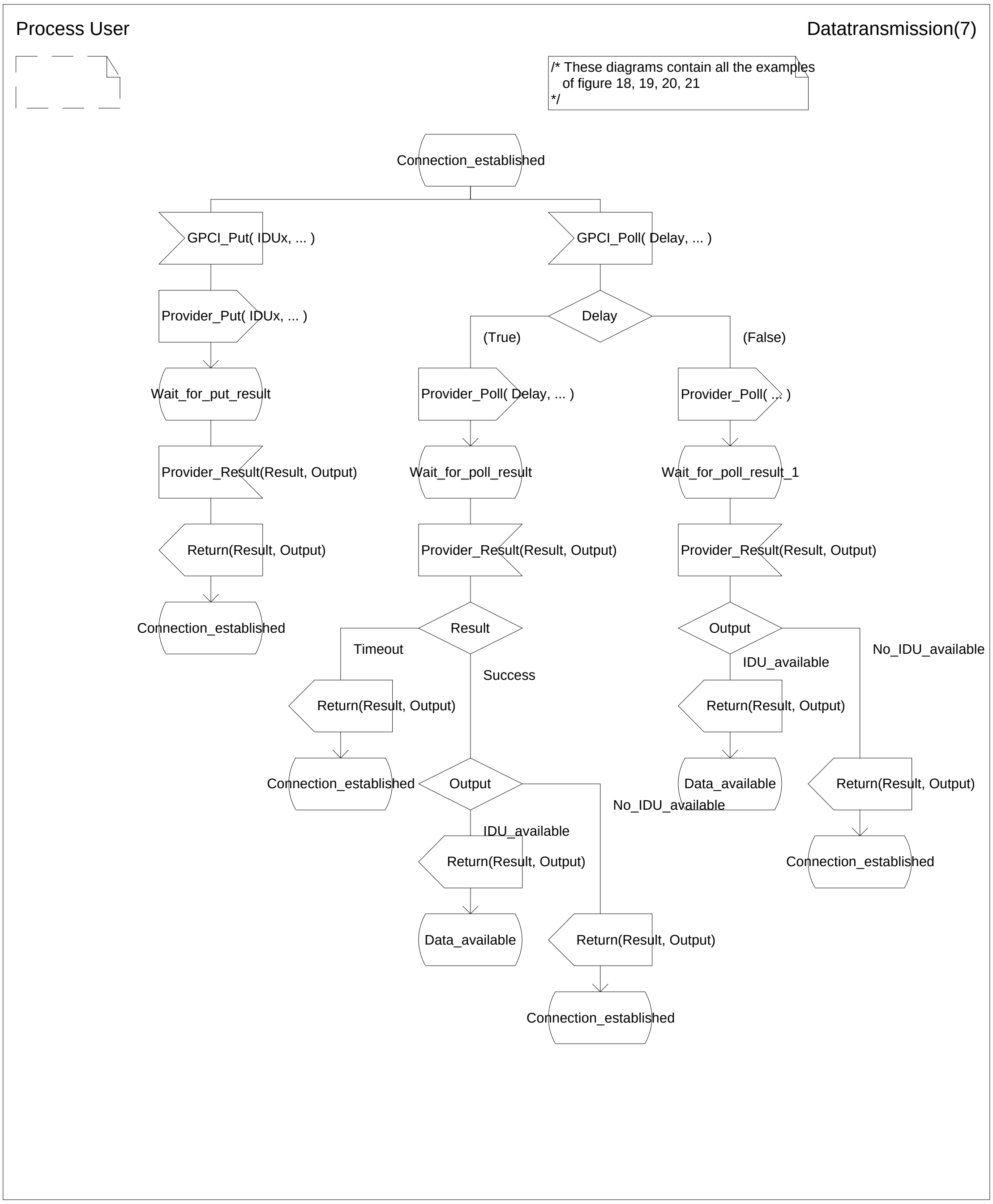

Process User
Datatransmission(7)
/* These diagrams contain all the examples
of figure 18, 19, 20, 21
*/
Connection_established
GPCI_Put(IDUx, ...)
GPCI_Poll(Delay, ...)
Provider_Put(IDUx, ...)
Delay
(True)
(False)
Wait_for_put_result
Provider_Poll(Delay, ...)
Provider_Poll(...)
Provider_Result(Result, Output)
Wait_for_poll_result
Wait_for_poll_result_1
Return(Result, Output)
Provider_Result(Result, Output)
Provider_Result(Result, Output)
Connection_established
Result
Output
Timeout
Success
No_IDU_available
IDU_available
Return(Result, Output)
Return(Result, Output)
Connection_established
Output
Data_available
Return(Result, Output)
No_IDU_available
IDU_available
Return(Result, Output)
Connection_established
Data_available
Return(Result, Output)
Connection_established

Figure B.5

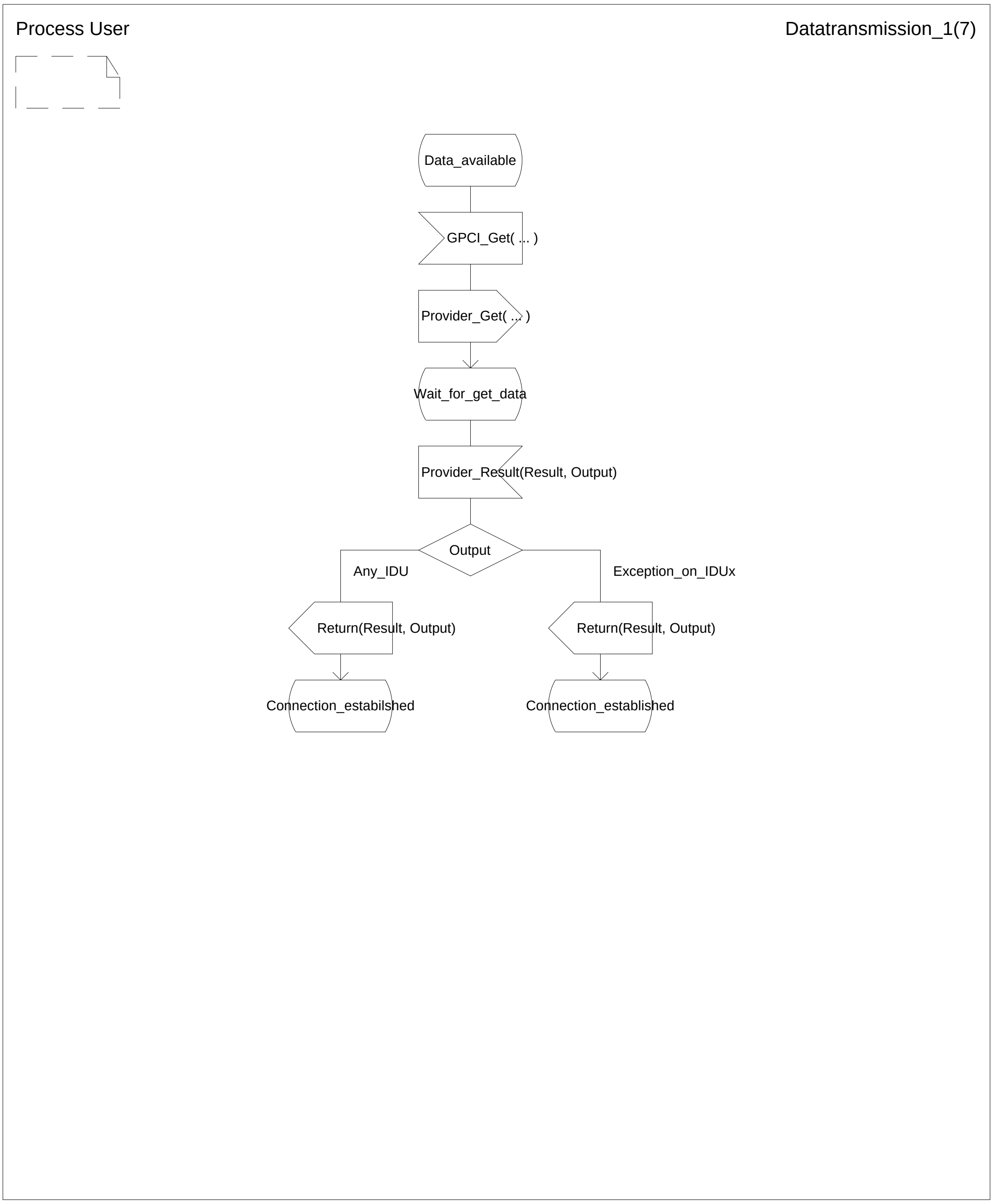

Process User
Datatransmission_1(7)
Data_available
GPCI_Get(...)
Provider_Get(...)
Wait_for_get_data
Provider_Result(Result, Output)
Output
Any_IDU
Exception_on_IDUx
Return(Result, Output)
Return(Result, Output)
Connection_estabilshed
Connection_established

Figure B.6

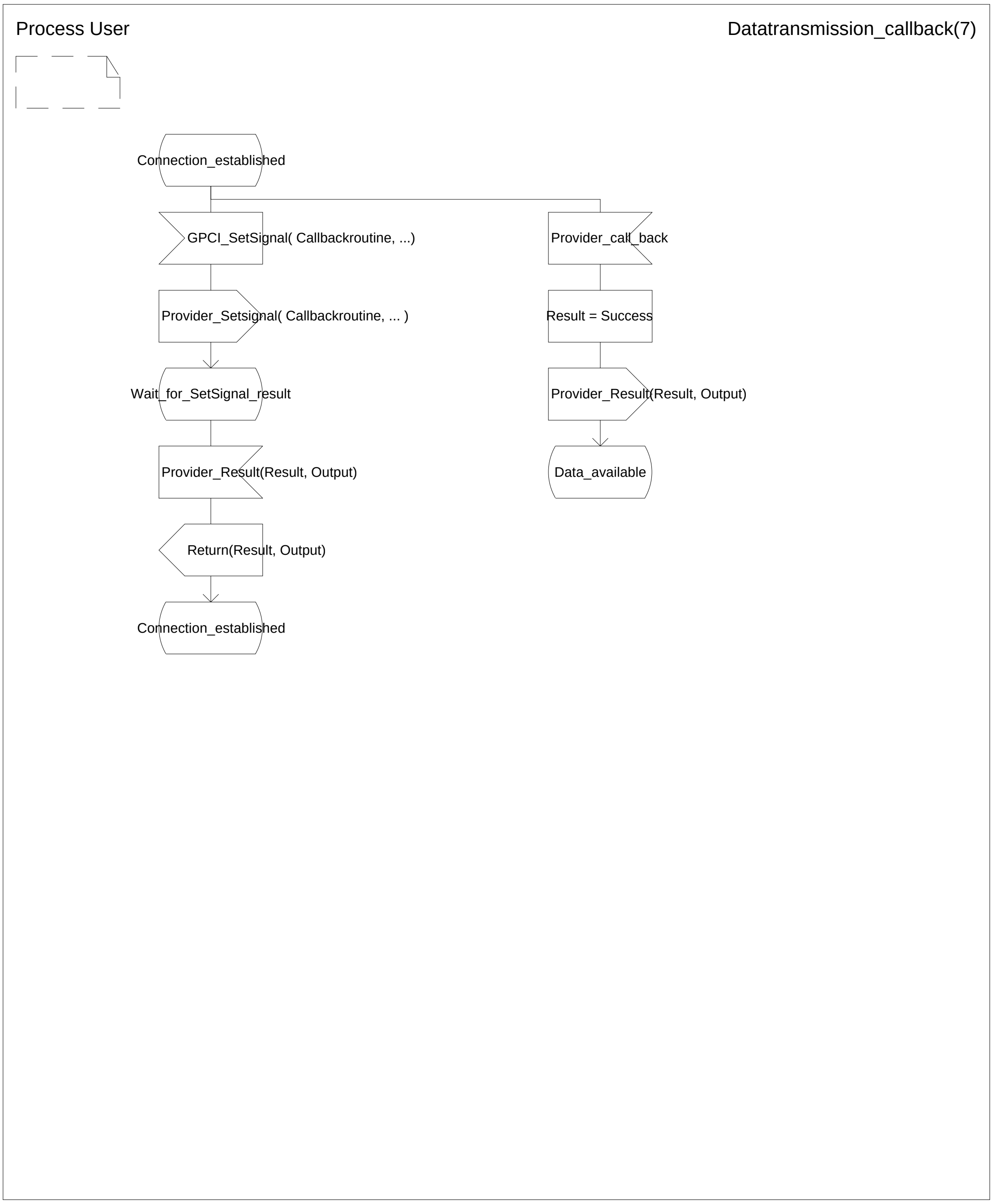

Process User
Datatransmission_callback(7)
Connection_established
GPCI_SetSignal(Callbackroutine, ...)
Provider_Setsignal(Callbackroutine, ...)
Wait_for_SetSignal_result
Provider_Result(Result, Output)
Return(Result, Output)
Connection_established
Provider_call_back
Result = Success
Provider_Result(Result, Output)
Data_available

Figure B.7

Annex C: Bibliography

The following documents were used as background material in the preparation of this ETR.

- ISO/IEC JTC1 N2890 (1994): "JTC 1 Plenary Meeting Report of Ad Hoc Group B on Application Program Interfaces".

- ITU-T Recommendation X.25: "Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to publication networks by dedicated circuit".

- ITU-T Recommendation T.611 (1994): "Programmable communication interface (PCI) APPLI/COM for facsimile Group 3, facsimile Group 4, teletex telex, E-Mail and file transfer services".

- ETS 300 243-1 (1995): "Terminal Equipment (TE); Programmable Communication Interface (PCI) APPLI/COM for facsimile Group 3, facsimile Group 4, teletex and telex services, Part 1: ITU-T Recommendation T.611 (1992) [modified]".

- ETS 300 243-2 (1995): "Terminal Equipment (TE); Programmable Communication Interface (PCI) APPLI/COM for facsimile Group 3, facsimile Group 4, teletex and telex services, Part 2: Conformance testing".

- ETS 300 325 (1994): "Integrated Services Digital Network (ISDN); Programmable Communication Interface (PCI) for Euro-ISDN".

History

Document history	
March 1997	First Edition

ISBN 2-7437-1309-7
Dépôt légal : Mars 1997