Interoperability Specification for ICCs and Personal Computer Systems

Part 1. Introduction and Architecture Overview

Apple Computer, Inc.

Axalto

Gemplus SA

Infineon Technologies AG

Ingenico SA

Microsoft Corporation

Philips Semiconductors

Toshiba Corporation

Revision 2.01.01

September 2005

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE.
NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED OR INTENDED HEREBY.
AXALTO, BULL CP8, GEMPLUS, HEWLETT-PACKARD, IBM, MICROSOFT, SIEMENS NIXDORF, SUN MICROSYSTEMS, TOSHIBA AND VERIFONE DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF PROPRIETARY RIGHTS, RELATING TO IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. AXALTO, BULL CP8, GEMPLUS, HEWLETT-PACKARD, IBM, MICROSOFT, SIEMENS NIXDORF, SUN MICROSYSTEMS, TOSHIBA AND VERIFONE DO NOT WARRANT OR REPRESENT THAT SUCH IMPLEMENTATION(S) WILL NOT INFRINGE SUCH RIGHTS.

Revision History

Revision	Issue Date	Comments
2.00.11	May 28, 2004	Spec 2.00 Final Draft
2.01.00	June 23, 2005	Spec 2.01 Final Release
2.01.01	September 29, 2005	Changed Schlumberger to Axalto

Contents

1 Introduction

The PC/SC Workgroup was initiated to develop a specification that can facilitate the interoperability necessary to allow Integrated Circuit Card (ICC) technology to be effectively utilized in the PC environment. In addition to development of the specification, the PC/SC Workgroup members are committed to implementation of both hardware devices and PC system components necessary to validate the design efforts. This is deemed a critical step in the process of moving toward accepted standards and will provide a base of experience from which to further refine and/or enhance this specification.

The PC/SC Workgroup will retain ownership of this specification until such time as it can be submitted and accepted by a formal standards body. The Workgroup will work with other interested parties to make this happen as quickly as possible. Until that time, the PC/SC Workgroup will support the general review by the PC and ICC communities at large, and evolution of the specification as necessary to respond to this general review process.

This specification is available on a royalty-free basis to any party wishing to implement a compliant product.

1.1 Motivation

The Integrated Circuit Card (ICC) is an intrinsically secure computing platform ideally suited to providing enhanced security and privacy functionality for applications running within general purpose computing environments such as the personal computer (PC). ICCs are capable of providing secure storage facilities for sensitive information such as:

- Private keys.
- Account numbers.
- Passwords.
- Medical information.

At the same time, the ICC provides an isolated processing facility capable of using this information without exposing it within the PC environment where it is at potential risk from hostile code (viruses, Trojan horses, and so on). This becomes critically important for certain operations such as:

- Generation of digital signatures, using private keys, for personal identification.
- Network authentication based on stored secrets.
- Maintenance of electronic representations of value (that is, prepaid purchase credits).

Currently, the use of ICCs in the PC environment is hampered by the lack of interoperability at several levels. First, the industry lacks standards for interfacing PCs to IFDs. This has made it difficult to create applications that can work with IFDs from a variety of vendors. Attempts to solve this problem in the application domain invariably increase costs for both development and maintenance. It also creates a significant problem for the PC user in that an IFD used with one application may not work with future applications.

Second, there is no widely accepted high-level programming interface for common ICC functionality. Encapsulation of ICC interfaces can dramatically simplify application development and reduce costs by allowing low-level interface software to be shared across multiple applications. In addition, a standardized high-level interface allows applications to reduce their dependency on a specific ICC implementation, making it far more likely that an application will be able to use future, enhanced ICCs. To increase the flexibility and independence of PC-based software components necessary to deal with multi-application ICCs, it is also important to reflect the different roles of ICC vendors and ICC issuers through the mechanisms provided for ICC-application deployment. Only by separating ICC technologies from the ICC-applications can specific ICC technologies become transparent to ICC-based PC applications when dealing with ICC-applications.Third, mechanisms to allow multiple applications to effectively share the resources of a single ICC are not defined. These are critically important as we rapidly move toward the deployment of multiple-application ICCs and generic cryptographic ICCs that will be used as part of a multiprocessing PC environment. Without agreed upon standards for device sharing, it becomes effectively impossible for application developers to ensure that they can complete an operation using ICC services without interruption.

To optimize the benefit to both the industry and end users, it is important that solutions to these issues be developed in a manner that supports a variety of operating environments and a broad base of applications. Only through this approach can we support the needs of all constituencies and encourage development of ICC-based PC applications, as a cost-effective solution to meeting requirements in a very diverse set of markets.

ICC technology offers a vital addition to the security infrastructure of the PC and network environments. It is an enabling technology for network commerce in general. To achieve this potential, however, it is essential that a consistent framework exists into which the diverse efforts of application developers, network technology vendors, and ICC technology vendors can be coherently channeled. Establishing this framework will enhance the applications and services available to the end consumer, enhance the marketplace for application developers, enhance the marketplace for network technology vendors, and enhance the marketplace for ICC technology vendors. Establishing this framework is the basic motivation of the PC/SC Workgroup's efforts.

1.2 Objectives

This document provides an overview of the specification, describing the minimum functionality required of ICCs, ICC Interface Devices (IFDs), and PCs to allow interoperability among compliant elements as provided by a variety of vendors.

The specification as a whole seeks to achieve the following objectives:

- Maintain consistency with existing ICC-related and PC-related standards while expanding upon them where necessary and practical.
- Enable interoperability among components running on various platforms (platform neutral).
- Enable applications to take advantage of products and components from multiple manufacturers (vendor neutral).
- Enable the use of advances in technology without rewriting application-level software (application neutral).

- Facilitate the development of standards for application-level interfaces to ICC services in order to enhance the fielding of a broad range of ICC-based applications in the PC environment.
- Support an environment that encourages the widest possible use of ICCs as an adjunct to the PC environment.

1.3 History of the PC/SC Workgroup

At the time of release of revision 1.0 of the specification in December 1997, the PC/SC Workgroup was composed of Bull CP8, Gemplus, Hewlett-Packard Corporation, IBM Corporation, Microsoft Corporation, Axalto, Siemens Nixdorf, Sun Microsystems, Toshiba, and Verifone.

At the time of publication of Revision 2.0 of this specification, the PC/SC Workgroup now consists of the following core member companies: Apple, Axalto, Gemplus, Infineon Technologies, Ingenico, Microsoft Corporation, Philips Semiconductors, and Toshiba. The PC/SC Workgroup has also brought a large number of associate member companies into the group.

1.4 Organization of this Specification

The *Interoperability Specification for ICCs and Personal Computer Systems* is composed of nine parts. These are intended to apply only to devices and software intended to operate as a part of an overall system that includes a personal computer. Their potential application in other environments is outside the scope of this specification.

The Parts of this specification detail specific interoperability requirements for compliant devices, reference design information, programming interfaces, and functional compatibility requirements. These documents include:

- Part 1. Introduction and Architecture Overview
- Part 2. Interface Requirements for Compatible IC Cards and Interface Devices
- Part 3. Requirements for PC-Connected Interface Devices
- Part 4. IFD Design Considerations and Reference Design Information. This part is here as an example reference only.
- Part 5. ICC Resource Manager Definition
- Part 6. ICC Service Provider Interface Definition
- Part 7. Application Domain/Developer Design Considerations
- Part 8. Recommendation for Implementation of Security and Privacy ICC Devices
- Part 9. IFDs with Extended Capabilities

This document is Part 1, “Introduction and Architecture Overview.” It discusses the overall purpose and goals of the specification, provides references to related material, and defines the smart card terminology used in this specification. It also presents the components and architectural assumptions underlying this specification.

1.5 Target Audience

This specification is intended for the following users:

- Operating system developers who wish to support ICCs as standard peripheral devices
- OEMs interested in developing Interface Device peripherals
- OEMs interested in developing ICC products for use with PCs
- Application developers who wish to create ICC-based products

1.6 Related Documents

1.6.1 ICC-Related Standards

- ISO/IEC 7816 - 1 Identification Cards - Integrated Circuit(s) cards with contacts - Part 1: Physical characteristics
- ISO/IEC 7816 - 2 Identification Cards - Integrated Circuit(s) cards with contacts - Part 2: Dimensions and location of the contacts
- ISO/IEC 7816 -3 Identification Cards - Integrated Circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols
- ISO/IEC 7816 -3/1 Identification Cards - Integrated Circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols AMENDMENT 1: Protocol type T=1, asynchronous half duplex block transmission protocol
- ISO/IEC 7816 -3/2 Identification Cards - Integrated Circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols AMENDMENT 2: Revision of protocol type selection
- ISO/IEC 7816 -4 Identification Cards - Integrated Circuit(s) cards with contacts - Part 4: Inter-industry commands for interchange.
- ISO/IEC 7816 -5 Identification Cards - Integrated Circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers.
- ISO/IEC 7816 -6 Identification Cards - Integrated Circuit(s) cards with contacts - Part 6: Inter-industry data elements
- ISO/IEC 7816 -10 Identification Cards - Integrated Circuit(s) cards with contacts - Part 10: Electronic signals and answer to reset for synchronous cards
- ISO/IEC 7811 1995 Identification Cards [REFERENCE TO PHYSICAL SPECS IN 7811]
- ISO/IEC 14443 -1 Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Physical characteristics
- ISO/IEC 14443 -2 Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Radio frequency power and signal interface
- ISO/IEC 14443 -3 Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Initialization and anticollision
- ISO/IEC 14443 -4 Identification Cards – Contactless integrated circuit(s) cards – Proximity cards – Answer to Select and Transmission Protocol
- ISO/IEC 15693 – 1 Identification cards — Contactless integrated circuit(s) cards — Vicinity Integrated Circuit(s) Card – Physical Characteristics
- ISO/IEC 15693 – 2 Identification cards — Contactless integrated circuit(s) cards — Vicinity Cards – Air Interface and Initialization
- ISO/IEC 15693 – 3 Identification cards — Contactless integrated circuit(s) cards — Anticollision and Transmission Protocol

- EMV ‘96 Integrated Circuit Card Application Specification for Payment Systems, Version 3.1.1
- EMV ‘96 Integrated Circuit Card Specification for Payment Systems, Version 3.1.1
- EMV ‘96 Integrated Circuit Card Terminal Specification for Payment Systems, Version 3.1.1
- RFC 2279: F. Yergeau, “UTF-8, a transformation format of ISO 10646” (Obsoletes 2044), Alis Technologies, January 1998
- CEN prEN 726-1: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 1: System Overview
- CEN prEN 726-2: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 2: Security Framework
- CEN prEN 726-3: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 3: Application-Independent Card Requirements
- CEN prEN 726-4: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 4: Application-Independent Card-Related Terminal Requirements
- CEN prEN 726-5: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 5: Payment Methods
- CEN prEN 726-6: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 6: Telecommunication Features
- CEN prEN 726-7: Identification Card Systems - Telecommunications Integrated Circuits Cards and Terminals - Part 7: Security Module

1.6.2 Personal Computer Interface Specifications

- Microsoft Press® Computer Dictionary, 1991, Microsoft Press
- Hardware Design Guide for Microsoft® Windows® 95, 1994, Microsoft Press
- EIA RS-232C, Electronics Industries Association, Washington, D.C.
- Shanley Tom, ISA System Architecture, 1993, Mindshare, Inc.
- PC Card Standard, Volume 5. Card Services Specification, January 1995, PCMCIA/JEIDA
- IBM-AT Technical Reference Manual, First Edition, March 1984, International Business Machines (reprints may be ordered from www.annabooks.com)
- Microsoft Cryptographic API Application Programmer’s Guide
- Microsoft Cryptographic Service Provider Programmer’s Guide
- Universal Serial Bus Specification, Version 1.0 Draft Revision, Compaq, Digital Equipment Corporation, IBM PC Company, Intel, Microsoft, NEC, Northern Telecom, November 13, 1995

2 System Architecture and Components Overview

2.1 Architecture overview

The architecture defined by this specification, in terms of software and hardware components, is depicted in Figure 2-1.

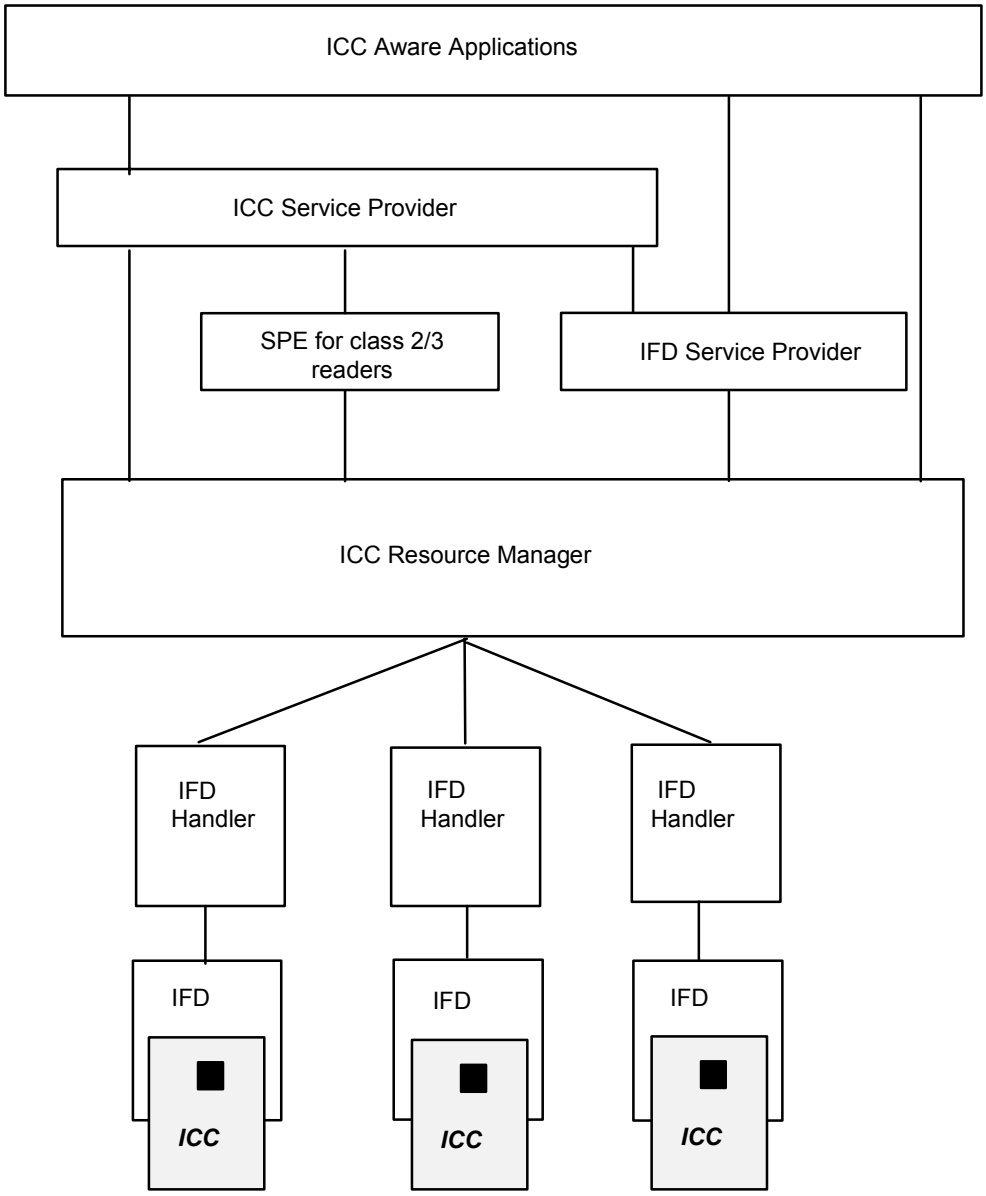

Figure 2-1. General Architecture

The elements that comprise the architecture are defined in the following sections.

2.1.1 Integrated Circuit Card (ICC)

The ICC (commonly called a “smart card”) is a credit card–sized plastic case with an embedded microprocessor chip. An ICC compliant with this interoperability specification will conform physically and electrically to the ISO 7816-1, 7816-2, and 7816-3 standards. ICCs compliant with the ISO 7816-10 draft specification for synchronous cards are also supported by this specification. In addition to asynchronous and synchronous contact ICCs, some contactless ICCs are supported by this specification, such as ISO/IEC 14443 for proximity ICC (PICC) and ISO/IEC 15693 for vicinity ICC (VICC) as well as similarly operating devices. These standards provide a detailed definition of the physical form factor and the electrical characteristics of a compliant ICC.The aim of this specification is to provide the same interfaces and look–and–feel at the application level for contactless and contact ICCs.

An ICC is an intrinsically secure computer platform that offers a variety of services, varying from simple secure data storage, to sophisticated cryptographic services, to an ICC-aware application.

2.1.2 Interface Device (IFD)

The IFD (commonly called a “smart card reader”) is the physical interface device through which ICCs communicate with a PC. The IFD provides DC power to the microprocessor chip (DC = direct current, as opposed to alternating current). Also, the IFD provides a clock signal, which is used to step the program counter of the microprocessor, as well as an I/O line through which digital information may be passed between the IFD and the ICC.

An IFD may have one or more slots to read ICCs and may also support some extended capabilities such as display or PINpad. Whereas the main goal of this document is to specify the communications between ICCs and PC applications, Part 9 of the specification provides the means to address these IFD extended capabilities.

An IFD may use a variety of physical access ports to the PC. Typically, these will be the keyboard port, a serial line port, a PC Card (PCMCIA) or the Universal Serial Bus (USBport).

Typically, a compliant IFD will conform to the ISO 7816-1, 7816-2, and 7816-3 standards. In addition, IFDs may support “Synchronous card type 1” and “Synchronous card type 2.” as defined in the ISO 7816-10-draft specification for synchronous cards, or the ISO/IEC 14443 or 15693 protocol for contactless cards. It is the responsibility of the IFD Subsystem to handle the protocols necessary for those ICCs and to map APDUs given by the application to the corresponding ICC commands.For contactless ICCs, implementation of the IFD subsystem (IFD and IFD Handler) must emulate basic functional requirements such as card insertion and removal events and ATR or Attributes, respectively (Part 3). IFD subsystems must also take care of the initialization, selection and communication processes with ICCs. IFDs may vary widely in their implementations, allowing vendors to make tradeoffs between intelligence embedded within the device itself and within the IFD Handler software within the PC. For the simplest devices, an IFD need provide little more than electrical connectivity and I/O signal passing between the ICC and the PC. In more complex configurations, an IFD may actually support the data link layer protocols defined in the ISO 7816-3 standards.

2.1.3 Interface Device Handler (IFD Handler)

The IFD Handler encompasses the PC software necessary to map the native capabilities of the IFD to the IFD Handler interface defined in Part 3 of this specification.

The IFD Handler is typically low-level software within the PC that supports the specific I/O channel used to connect the IFD to the PC and provides access to specific functionality of the IFD. The differences between “smart” IFDs and “dumb” IFDs are hidden at the IFD Handler API. This is the layer of the interoperability specification primarily responsible for facilitating the interoperability between different IFDs.

In order to manage Enhanced IFD capabilities at the IFD Subsystem level, another type of device must be introduced in addition to the Slot Logical Device. This device is called the Functional Logical Device.

2.1.3.1 Slot Logical Devices

The Slot Logical Device is the terminus (on the PC side) of the ICC communication protocols (ISO 7816-3, ISO 7816-10, or ISO/IEC 14443 / 15693). At the IFD Handler level, the API hides all distinctions between ICCs based on ISO protocol handling, whether synchronous or asynchronous.

2.1.3.2 Functional Logical Devices

Depending on flexibility and security, one or several Functional Logical Devices can be present. For example, they can be mapped directly to a physical sub–device such as a display or keyboard, or regrouped as a PINpad Functional Logical Device. An Application Context, as defined in Part 9, lists the requirements associated with a set of functions or security features present in an IFD.

The main role of the Functional Logical Devices is to enable the IFD Service Provider to lock independent types of functionality of the IFD via the Resource Manager. However, the Resource Manager is completely transparent to the communication between the IFDSP and the functional logical devices. Indeed, unlike the Slot Logical Device, the Functional Logical Device supports only a generic communication channel.

Moreover, as these Functional Logical Devices are only used by the IFDSP, the ICC aware application or ICC Service Provider is completely independent of the Functional Logical Device layout. This preserves the interoperability between IFDs with different Functional Logical Device layouts.

2.1.4 ICC Resource Manager

The ICC Resource Manager is a key component of the PC/SC Workgroup’s architecture. It is responsible for managing the other ICC-relevant resources within the system and for supporting controlled access to IFDs and, through them, individual ICCs. The ICC Resource Manager is assumed to be a system-level component of the architecture. It must be present and will most likely be provided by the operating system vendor. There should be only a single ICC Resource Manager within a given system.

The ICC Resource Manager solves three basic problems in managing access to multiple IFDs and ICCs.

First, it is responsible for identification and tracking of resources. This includes:

- Tracking installed IFDs and making this information accessible to other applications.
- Tracking known ICC types, along with their associated Service Providers and supported Interfaces, and making this information accessible to other applications.
- Tracking ICC insertion and removal events to maintain accurate information on available ICCs within the IFDs.

Second, it is responsible for controlling the allocation of IFDs and resources (and hence access to ICCs) across multiple applications. It does this by providing mechanisms for attaching to specific IFDs in shared or exclusive modes of operations.

Finally, it supports transaction primitives on access to services available within a given ICC. This is extremely important, as current ICCs are single-threaded devices, which often require execution of multiple commands to complete a single function. Transactions allow multiple commands to be executed without interruption, ensuring that intermediate state information is not corrupted.

2.1.5 Service Providers

The Service Providers are responsible for encapsulating functionality exposed by a specific ICC or IFD, and making it accessible through high-level programming interfaces. These interfaces may be enhanced and extended to meet the needs of specific application domains.

An important point to note is that this specification does not require a Service Provider to be a monolithic component running on a single PC. In particular, one can envision building a Service Provider as a client/server component. This would allow a server-side application developer to take advantage of the high-level interfaces and interoperability supported by this architecture.

2.1.5.1 ICC Service Provider (ICCSP)

An ICCSP is a Service Provider that interfaces ICC functionality.

Revision 2.0 of this specification introduces different types of ICCSPs: ICC Operating System Service Providers (ICCOSSP) interface functionality from a specific ICC Operating System, while Service Providers that interface a particular on-card application will now be called Application Domain Service Providers (ADSP).

These new types of service providers reflect the different roles of ICC vendor, ICC issuer, and on-card application provider. This is done by linking the ICC type to an ICC–OS implementation only. The ICC issuer then maintains the linking of the on-card applications and their specific off–card components (ADSPs). This requires a new mechanism to introduce ICCOSSPs and ADSPs to the system. This mechanism is based on a new card recognition system (see 2.1.5.1.3).

In the Revision 1.0 service provider introduction mechanism, which is now deprecated, an ICCSP needed to be introduced to the Resource Manager in order to be used. Typically, this was done through an ICC setup utility provided by the ICC vendor. This utility must provide four pieces of information about the card:

1. Its ATR string and a mask to use as an aid in identifying the ICC
2. An identifier for the Service Provider(s) that support the ICC
3. A list of ICC Interfaces supported by the ICC
4. A “friendly name” for the ICC; to be used in identifying the ICC to the user (in most cases, the user will supply this to the setup utility)

2.1.5.1.1 ICCOS Service Provider (ICCOSSP)

The ICCOS Service Provider encapsulates access to functionality from a specific ICC Operating System through high-level programming interfaces. Interfaces for commonly implemented file access and authentication services are defined by this specification. If an ICC implements these services, it shall make use of the defined interfaces. However, additional interfaces may be defined and implemented to meet domain-specific requirements.Typically, an ICCOSSP needs to be introduced to the Resource Manager, to map it to a particular ICCOS. There is a one–to–one relationship between ICCs and their ICCOSSP. This will then be used in the new card recognition system (See 2.1.5.1.3).

2.1.5.1.2 Application Domain Service Provider (ADSP) and ADSP Locator (ADSPL)

2.1.5.1.2.1 Application Domain Service Provider (ADSP)

The ADSP is a service provider intended to interface with on–card applications. This differentiates the ADSP from a regular ICCSP which interfaces to an ICC–type or ICCOS.

ADSPs support on–card applications in a flexible way that is compatible with multi–application card environments. Contrary to ICCSPs, as defined in Revision 1.0, which can only be mapped to the ICC using the ATR through the Resource Manager, the ADSP uses a dynamic assignment mechanism using an ADSP Locator (see 2.1.5.1.2.2), and a specific Card Recognition System (see 2.1.5.1.3). The ADSP Locator mechanism allows dynamic assignment of an ADSP. This in turn will allow multiple ADSPs to be available to off–card applications for the same ICC.

A modification to an on–card application only needs to be accompanied with an updated implementation of the corresponding ADSP.

2.1.5.1.2.2 ADSP Locator (ADSPL)

The static linking between ICC–Type and available ICCSPs, as performed by the Resource Manager (RM), is not possible in a multi–application card environment. The ADSP Locator is introduced to the Resource Manager to allow dynamic assignment of ICC Service Providers.

The ADSPL is itself a service provider (typically written by the ICC issuer) that is loaded by the Resource Manager. It allows the RM to provide off–card applications with:

- A way of listing on–card applications.
- A way of retrieving a reference to the appropriate ADSP implementation related to a chosen on–card application.

The provider of the ADSPL is responsible for specifying the way in which ADSPs should be introduced to the ADSPL. The design of the ADSPL could be such that it can retrieve an appropriate ADSP automatically when it is needed.

2.1.5.1.3 Card Recognition System

In general, ICCs are identified by the ATR String they present to the off–card system. The flexibility of this common card recognition method is limited, due to the restricted length and complexity of an ATR. Card issuers, manufacturers and card application developers need to share the available ATR. Complex multi–application cards require an enhanced card recognition mechanism to retrieve the basic information for identifying an ICC.

All information regarding the identification of an ICC must be available on the ICC itself. The identity information is stored in a ICC Info Structure (or “extended” ATR). The information can be placed, for example, in a file or applet – depending on the used ICC technology. The ICC must include a command in the ATR’s historical bytes, which can be used by the off–card system (Resource Manager) to retrieve the ICC Info Structure. Specifying a command within the historical bytes of an ICC is described in more detail in the ISO 7816–4 document.

When dealing with this type of enhanced ICC, the Resource Manager interprets the historical bytes of the ATR, sends the included command back to the ICC, and retrieves the ICC Info Structure. The information from this structure is used by the Resource Manager to identify the ICC.

Part 5 of this specification defines a number of fields that can be used in the ICC Info structure, including a reference to the ICCOS and a reference to an ADSPL.

When the ICC is inserted, the Resource Manager will retrieve the ICC Info, get the ADSPL reference from the ICC Info Structure and load the ADSPL. When appropriate, the Resource Manager will retrieve the list of on–card applications from the ADSPL.

The off-card application gets this list from the Resource Manager. It can then choose from this list the appropriate on-card application and load the corresponding ADSP to interact with the on–card application.

ICCs supporting the enhanced Card Recognition mechanism do not need to be introduced to the Resource Manager as ICC-Types anymore. For these cards, only the ICCOSSP and ADSPL are introduced to the Resource Manager.

2.1.5.1.4 Cryptographic Service Provider (CSP)

The Cryptographic Service Provider, as defined in Revision 1.0 of this specification, encapsulates access to a specific ICC cryptographic functionality through high-level programming interfaces. It should expose only cryptographic functions to PC applications. Other functionality should be implemented in other Service Providers.

In Revision 2.0, a CSP can be implemented as an ADSP for an ICC that comprises the new card recognition system, or as a traditional ICCSP if the ICC is using the deprecated introduction mechanism of Revision 1.0. The notion of CSP does not represent a different type of service provider technology; it is there only to segregate cryptographic functions to facilitate the handling of export/import regulations.

Interfaces are defined in this specification for general-purpose cryptographic services including:

- Key generation.
- Key management.
- Digital signatures.
- Hashing (or message digests).
- Bulk encryption services.
- Key import/export.

2.1.5.2 IFD Service Provider (IFDSP)

When extended IFD capabilities are available, they are presented to the ICC-aware application or ICC Service Provider through high level programming interfaces implemented in an IFD Service Provider (IFDSP). An IFDSP encapsulates access, and interfaces with an IFD's functionality in the same way an ICC Service Provider interfaces with an ICC's functionality.

For each Application Context (which defines some type of functionality), the IFDSP may provide different interfaces. The interface implementation by the IFDSP interacts with the implementation of the IFD Handler in a mode that is transparent to the Resource Manager. This specification defines interfaces for commonly implemented services. However, additional interfaces may be defined and implemented for specific capabilities.

Since an IFD Service Provider can be composed of modular components, the services associated with an IFD can evolve, as in an IFD with download capability. However, such download procedure is out of the scope of this specification.

Part 9 of the Specification describes four pre–defined services that can be exported by an IFDSP, the support of which depends on IFD capabilities and security requirements: Secure PIN, Display, User confirmation, and Generic user entry.

2.1.6 ICC-Aware Application

The ICC-Aware Application ("Application") is an arbitrary software program within the PC operating environment, which wants to make use of the functionality provided by one or more ICCs. It is assumed the Application is running as a process within a multi-user, multiprocess, multiple-threaded, and multiple device environment. The architecture components defined within this specification provide mechanisms to map PC application

requests to the ICC, which is typically a single user, single-threaded, but multiple application environment.

This overall architecture can alternatively be presented as a peer-to-peer communication protocol, as illustrated in the following figure.

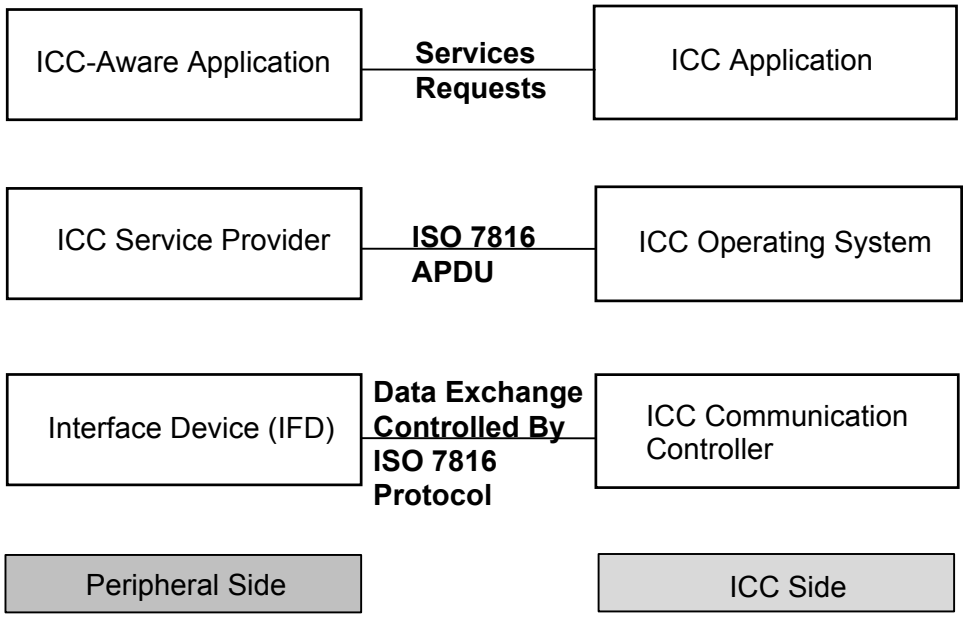

Figure 2-2. ICC/PC Communication Layers – the example shows ISO 7816

2.2 Specification Breakdown

The following illustration shows how the different parts that make up this specification can be related to the overall system architecture:

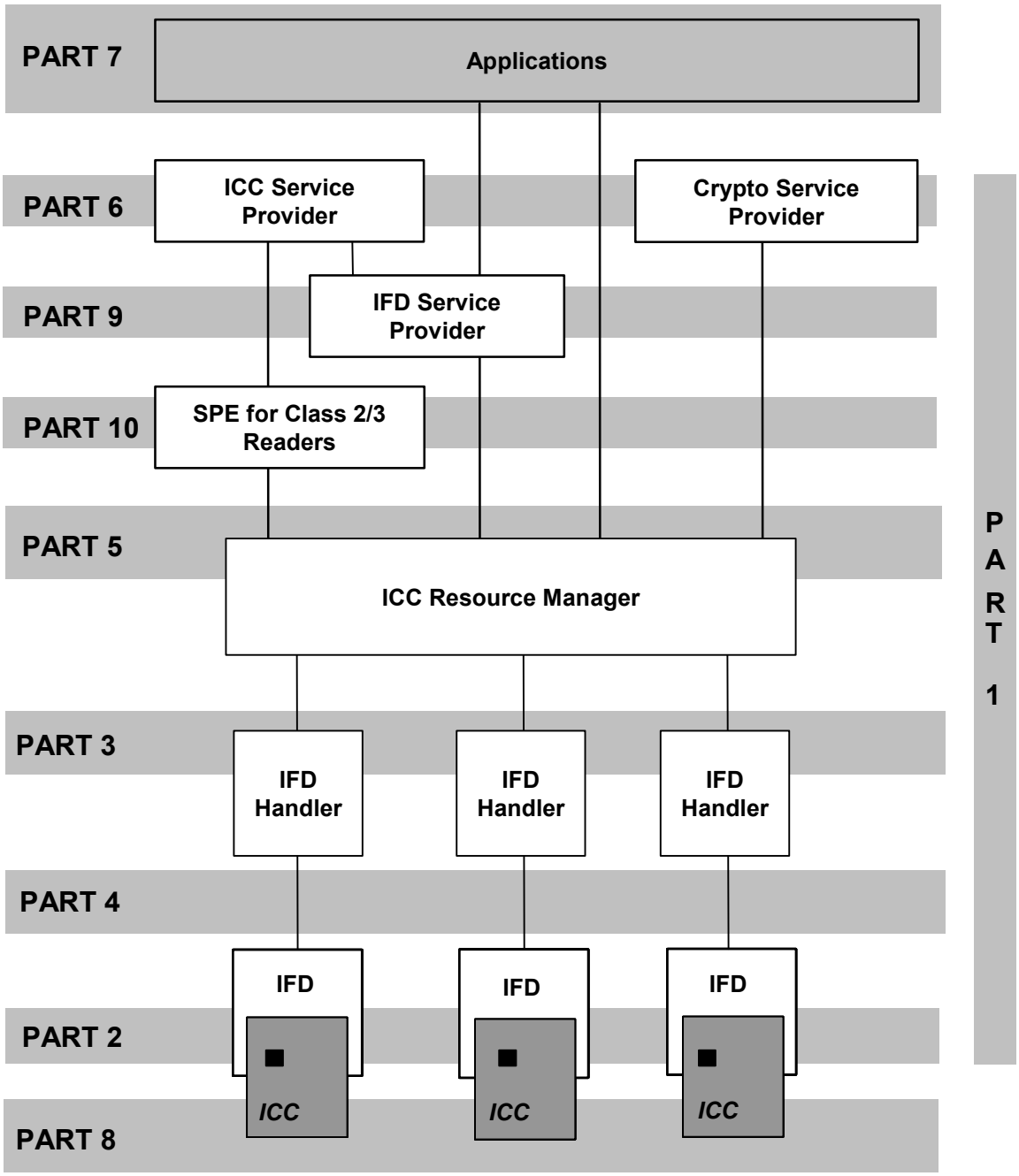

Figure 2-3. Specification Breakdown

2.3 System Requirements

In order to support the above-described architecture, the host PC system is assumed to provide:

- A general purpose Operating System environment that supports:
  - Multiple concurrent processes.
  - Process separation at the kernel level.
  - A shared library mechanism to facilitate code reuse and dynamic linking to shared code.
  - Asynchronous event/messaging support.
  - An interprocess communication facility.
  - A memory management facility.
- Support for 3rd party peripheral devices to include:
  - Installable peripheral devices.
  - The ability to support multiple devices of a given type.

2.4 Relevant Objects

Within the PC/SC Workgroup architecture, the following objects are recognized as being particularly relevant to ICC operations. A brief description is provided here because these terms are used within other Parts of this specification with a specific definition intended.

2.4.1 System

The “system” is the computer and connected peripheral devices. These specifications focus on personal computers with a single user; however, they can be extended to multi-user, multiple terminal computers.

2.4.2 User

The “user” is the end user of the PC system. In general, a user is identified and authenticated based on an Operating System–defined mechanism. Processes started on behalf of the user are generally deemed to be running on the user’s behalf and have the access rights and/or security attributes associated with the user. In terms of this specification, the user is also the ICC cardholder.

2.4.3 Process

This document uses the term “process” in its larger “computer science” definition. It refers to executing software along with its associated data and the security context under which it is running. Typically, a process is associated with the user (except for certain OS processes) and takes on the permissions and privileges of the user as determined by the Operating System.

2.4.4 Terminal

A “terminal” in this context consists of a station through which the user interacts with the PC. It normally consists of the primary user interface elements (monitor, keyboard,

mouse, and so on) and co-located peripherals, such as IFDs. Multiple processes may be associated with a single terminal, but at most one process controls the terminal at any given time.

2.4.5 ICC

ICCs are intelligent devices that may be used as PC peripheral devices. They will generally be “owned” by an individual, the cardholder, though they might be owned by a group. It is generally assumed these are high-value devices, which will remain in the possession of the cardholder when not required by an application.

In this specification, it is assumed that the cardholder is responsible for physical protection of the asset, but that the ICC itself will provide logical protection to its internal services.

2.4.6 IFD

IFDs are assumed to be peripheral devices on the PC making use of standard I/O channels. They may be system wide or associated with a specific terminal. Most existing PC devices support only a single user. However, this specification is intended to be equally usable on multi-user computers including existing Unix workstations.

The PC/SC Workgroup architecture supports flexible mechanisms for managing peripheral devices and managing connections between applications and these devices, to address both single user and multi-user environments.

3 Definitions, Abbreviations, and Symbols

AAC Application Authentication Cryptogram.

AC Access conditions. A set of security attributes associated with a file in a card.

AC Application Cryptogram.

ACK Acknowledgment.

Application Domain Service Provider (ADSP) A service provider for a specific on–card application.

ADSP Locator (ADSPL) A service provider that allows enumeration of on–card applications and assignment of an appropriate ADSP.

AID Application Identifier. A data element that identifies an application in a card. An application identifier may contain a registered application provider number. If it contains no application provider number, then this identification may be ambiguous.

ALW Access condition indicating a given function is always accessible

ANSI American National Standards Institute.

APDU Application protocol data unit.

Application The implementation of a well-defined and related set of functions that perform useful work on behalf of the user. It may consist of software and or hardware elements and associated user interfaces.

Application Context An identifier for a list of requirements associated with a set of functions and security features present in an IFD.

Application provider An entity that provides an application.

ARPC Authorization Response Cryptogram.

ARQC Authorization Request Cryptogram.

ASC Application Specific Command set. A DF can be associated, optionally, with an ASC. This means that when selecting this application, the general command set is extended or modified by this specific command set. The ASC is valid for the whole subtree of this application unless there are other ASCs defined at the lower level of this application.

ASN.1 object Abstract Syntax Notation object as defined in ISO/IEC 8824.

ATC Application Transaction Counter.

ATR Answer-to-Reset. The transmission sent by an ICC to the reader in response to a RESET condition.

ATS Answer to Select. The transmission sent by an PICC type A to the PCD in response to a SELECT condition

ATQA Answer to Request for PICC type A. This is sent by the PICC type A back to the PCD upon a REQA command.

ATQB Answer to Request. The transmission sent by an PICC type B to the PCD in response to a REQUEST condition

AUT Authenticated.

BCD Binary-coded decimal.

BGT Block Guard Time.

Block Logically contiguous data memory that is allocated when requested for data field.

BWI Block Waiting Time Integer.

BWT Block waiting Time.

Byte Eight bits.

C-APDU Command APDU.

CEN Comité Européen de Normalisation or European Committee for Standardization.

Certification Authority Trusted third party that establishes a proof that links a public key and other relevant information to its owner

CHV Card Holder Verification.

CID Card IDentifier. Number assigned to active PICCs that allows concurrent access to several cards from a single PCD

CLA Class Byte of the Command Message.

CLK Clock.

Cold reset The reset of an ICC that occurs when the supply voltage (VCC) and other signals to the ICC are raised from the inactive state and the reset (RST) signal is applied.

Command A message sent by the terminal to the ICC that initiates an action and solicits a response from the ICC.

Command/response pair Set of two messages: a command following by a response.

Contact A conducting element ensuring galvanic continuity between integrated circuit(s) and the external interfacing equipment.

Cryptogram Result of a cryptographic operation.

CWI Character Waiting Time Integer.

CWT Character Waiting Time.**DAD** Destination Node Address.

CRC Cyclic Redundancy Check

Data unit The smallest set of bits that can be unambiguously referenced.

DDK Device Driver Kit.

DEA Data Encryption Algorithm (ANSI X3.92). Public sector standard for DES.

DES Data Encryption Standard (FIPS Pub 46).

DF Dedicated file. File containing file control information, and, optionally, memory available for allocation. It may be the parent of elementary files and/or dedicated files.

DF name String of bytes that uniquely identifies a DF in the card.

DIR file Directory file. An optional elementary file containing a list of applications supported by the card and optional related data elements defined in ISO 7816/5.

DSA Digital Signature Algorithm. A cryptographic algorithm used in generating digital signatures as specified in FIPS 180-1.

DSS Digital Signature System. A procedure for generating digital signatures as specified in FIPS 180-1.

EDC Error Detection Code.

EF Elementary file. A set of data units or records which share the same identifier. It cannot be a parent of another file.

Enhanced IFD An IFD with extended capabilities.

Error condition Condition including rejection of a command.

ETU Elementary Time Unit. The nominal bit duration used in communication between an ICC and a reader. Defined in ISO 7816-3, paragraph 6.1.1.

FCI File Control Information.

File control information Logical, structural, and security attributes of a file as defined in ISO/IEC 7816.

File identifier A 2-byte binary value used to address a file

FMD File management data.

Function A process accomplished by one or more commands and resultant actions that are used to perform all or part of a transaction.

Functional Logical Device A physical part, a group of physical parts, or a functionality of an IFD exposed as a logical device. This is at the same level as the slot logical device.

GND Ground.

Guardtime The minimum time between the trailing edge of the parity bit of a character and the leading edge of the start bit of the following character sent in the same direction.

GUI Graphical User Interface.

Half-duplex transmission Two-way electronic communication that takes place in only one direction at a time. Communication between people is usually half-duplex—one listens while the other speaks (Source: *Microsoft Press® Computer Dictionary*).

I-block Information block associated with the T=1 protocol and the 14443 transmission protocol.

ICC Integrated Circuit Card. In this specification, used to refer to a plastic card containing an integrated circuit, which is compatible with ISO 7816.

ICC Info Structure A data structure stored on the ICC. It contains the information required for the enhanced card recognition mechanism.

ICC Label A friendly name associated with the ICC .This label is in the UTF-8 format, according to RFC-2279.

ICCOS The operating system of an ICC.

ICCOS-REFID An identifier associated with the ICC OS. It is a part of the ICC Info Structure.

ICCOSSP A service provider that interfaces functionality from a specific ICC Operating System

ICCSP A service provider that interfaces ICC functionality. ICCOSSPs and ADSPs are ICCSPs.

ICC-Type The specific implementation of the ICC technology.

IEC International Electrotechnical Commission.

IFD Interface Device. A terminal, communication device, or machine to which the integrated circuit(s) card is electrically connected during operation. As used in this specification, refers to a PC peripheral device that supports bidirectional I/O to an ISO 7816 standard ICC.

IFD Service Provider (IFDSP) A service provider that interfaces an IFD's extended capabilities.

IFD Subsystem The IFD, I/O channel, I/O device driver and IFD handler on the PC side.

IFS Information Field Size associated with the T=1 protocol.

IFSC Information Field Size for the ICC associated with the T=1 protocol.

IFSD Information Field Size for the terminal associated with the T=1 protocol.

IFSI Information Field Size Integer associated with the T=1 protocol.

Inactive The supply voltage (VCC) and other signals to the ICC are in the inactive state when they are at a potential of 0.4 V or less with respect to ground (GND).

INF Information field associated with the T=1 protocol.

INS Instruction Byte of Command Message associated with the T=0 and T=1 protocol.

Integrated Circuit(s) Electronic component(s) designed to perform processing and/or memory functions.

Internal Elementary File Elementary file for storing data interpreted by the card.

ISO International Organization for Standardization

Key qualifier An unambiguous set of data to select a specific keyset in the SM.

Key Pad An arrangement of numeric, command, and potentially function and/or alphanumeric keys laid out in an ordered manner.

LEN Length.

Level Number of DFs in the path to a file, starting the path from the MF.

LRC Longitudinal Redundancy Check associated with the T=1 protocol.

MCT Multifunctional Card Terminal

MD5 A one-way hash, or message digest, function.

Message String of bytes transmitted by the internal device to the card or vice versa, excluding transmission-control characters.

MF Master file. Mandatory unique dedicated file representing the root of the structure.

NAD Node address associated with the T=1 protocol.

NAK Negative ACK.

NEV An access condition indicating a given function is never accessible.

Nibble Half a byte. The most significant nibble of a byte consists of bits b_8 b_7 b_6 b_5 and the least significant of bits b_4 b_3 b_2 b_1.

P1(2) Parameters used in the T=0 and T=1 protocol.

Parent file The MF or DF immediately preceding a given file within the hierarchy.

Password Data that may be required by the application to be presented to the card by its user before data can be processed.

Path Concatenation of file identifiers without delimitation. If the path starts with the MF identifier, it is an absolute path.

PC Personal computer. For purposes of this document, PC refers to any host computer to which the ICC reader is attached.

PCB Protocol Control Byte.

PCD Proximity Coupling Device: ISO/IEC 14443 Contactless Reader.

PICC Proximity Integrated Circuit(s) Card. Contactless cards operating without mechanical contact to the IFD, using magnetic coupling.

PIN Personal Identification Number.

Pin Pad An arrangement of alphanumeric and command keys to be used for PIN entry.

PIX Proprietary Application Identifier Extension.

Plug and Play (PnP) A technology that detects the presence of hardware and directs the automatic loading of software drivers for that hardware. It also detects changes in the hardware set and directs drivers to load or unload as a result. (Source: MSDN Development Library)

Provider Authority who has or who obtained the rights to create the MF or a DF in the card.

PUPI Pseudo-Unique PICC Identifier

PTS Protocol Type Selection.

RATS Request for Answer to Select. Command sent by the PCD to a type A PICC to obtain its ATS.

R-APDU Response APDU.

R-block Receive Ready Block.

RF Radio Frequency

ICC Reader (or Reader) As used in this specification, refers to the slot of a PC peripheral device that supports bidirectional I/O to an ISO 7816 standard ICC. An IFD may have several slots and therefore be composed of several readers. It is associated with a “ReaderName”.

Record String of bytes that can be handled as a whole by the card and referenced by a record number or by a record identifier.

Record Identifier Value associated with a record that can be used to reference that record. Several records may have the same record identifier within an EF.

Record number A sequential number assigned to each record that uniquely identifies the record within its EF.

REQA Request Command, Type A. Command sent by a PCD to probe its energizing field for type A PICC presence.

REQB Request Command, Type B. Command sent by a PCD to probe its energizing field for type B PICC presence.

Response A message returned by the ICC to the terminal after the processing of a command message received by the ICC.

RFU Reserved for Future Use.

RID Registered application provider identifier.

RSA Rivest, Shamir, and Adleman Public-Key Cryptography.

RST Reset.

R-TPDU Response TPDU.

SAD Source Node address associated with the T=1 protocol.

SAK Select Acknowledge.

S-block Supervisory Block.

Script A command or a string of commands transmitted by the issuer to the terminal for the purpose of being sent serially to the ICC as commands.

Secrets Algorithms, related keys, security procedures, and information.

SFI Short File Identifier.

SHA-1 Secure Hash Algorithm Rev 1. A one-way hash, or message digest algorithm.

Slot See Reader

Slot Logical Device The terminus (on the PC side) of the ICC communication protocols (ISO 7816-3, ISO 7816-10, or ISO 14443).

SM Secure Messaging.

SM Security Module. A device containing logically and physically protected secrets in such a way that unauthorized access is not feasible. In order to achieve this the module may in addition be further physically, electrically, and logically protected.

SPE Secure PIN Entry

SPK Secure Presence Key. An isolated key implemented as part of an IFD subsystem. It is designed to provide a user confirmation signal to an ICC.

State A Space (as defined in ISO 1177).

State H High state logic level.

State L Low state logic level.

State Z Mark (as defined in ISO 1177).

SW1 (2) Status Byte 1 (2).

T=0 Character-oriented asynchronous half duplex transmission protocol.

T=1 Block-oriented asynchronous half duplex transmission protocol.

TAL Terminal Application Layer.

TC Transaction Certificate.

TCK Check Character.

TDOL Transaction Certificate Data Object List.

Template A grouping of data objects based upon their location within application structures that gives the context within which the data objects are to be interpreted. The template is given a name (a tag), which is used to reference the context and is also used as the tag for a constructed data object within which the data objects may appear in the value field.

TLV Tag-Length-Value.

TPDU Transport Protocol Data Unit.

TTL Terminal Transport Layer.

UID Unique IDentification

VCC Supply Voltage.

VCD Vicinity Coupling Device: ISO/IEC 15693 Read/Write Device.

VICC Vicinity Cards according to ISO/IEC 15693; Vicinity Integrated Circuit(s) Cards (VICC): Contactless ICCs which operate without mechanical contact to the VCD using magnetic coupling.**VPP** Programming Voltage.

Warm reset The reset of an ICC that occurs when the supply voltage (V_{CC}) and the clock (CLK) lines are maintained in their active state and the reset (RST) signal is applied.

Warning condition Indication that an error may have occurred when processing a command.

WI Waiting Time Integer.

Working elementary file Elementary file for storing data not interpreted by the card.

WTX Waiting Time Extension.