WP31 - Interface Device Service

D31.1 - Requirements report

Document Identification	
Date	28.02.2013
Status	Final
Version	1.0

Related SP / WP	SP3 / WP31	**Document Reference**	D31.1
Related Deliverable(s)	D22.x	**Dissemination Level**	PU
Lead Participant	TUD	**Lead Author**	Moritz Horsch
Contributors	Moritz Horsch (TUD), Pouyan Sepehrdad (TUD), Christoph Busold (TUD), Johannes Schmölz (ECS), Frank-Michael Kamm (G&D), Christian Wagner (TUG)	**Reviewers**	Christian Kahlo (AG), Eray Özmü (USTUTT), Chritopher Ruff (USTUTT)

This document is issued within the frame and for the purpose of the FutureID project. This project has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424

Abstract

The IFD service provides a common interface for communication to various devices like smart cards and secure elements. Applications are able to access such devices without considering the particular interface. This provides platform independency and interoperability for applications.

The IFD service of the FutureID client should provide a generalized interface for smart cards and card terminals. It must support PC/SC [1], NFC [2] [3] and SICCT [4] to connect to such devices. To secure the communication between the IFD and a device, for instance a smart card, the IFD must provide an interface to implement protocols through a trusted channel can be established. Additionally, the IFD should consider secure platforms especially the Open Mobile API [5] to integrate secure elements and the Trusted Platform Module (TPM) [6]. The IFD service must support ISO/IEC 27427-4 [7] and allow different bindings like SOAP [8] and PAOS [9]. Furthermore, the IFD should serve as a proxy to use multiple interfaces so that various devices can be accessed simultaneously.

Document Information

History

Version	Date	Author	Changes
0.1	19.11.2012	Moritz Horsch	1st Draft
0.2	05.12.2012	Moritz Horsch	Added introduction
0.3	06.12.2012	Moritz Horsch	Added sections 2 and 2.1
0.4	07.12.2012	Johannes Schmölz	Added sections 2.3 and 2.4
0.5	17.12.2012	Frank-Michael Kamm	Added section 2.2
0.6	08.01.2013	Moritz Horsch	Added abstract
0.7	10.01.2013	Moritz Horsch	Added conclusion
0.8	17.01.2013	Christian Wagner	Added section 2.5
0.9	17.01.2013	Pouyan Sepehrdad Christoph Busold	Fixed some typos and contents
0.91	17.01.2013	Moritz Horsch	Finalized version for review
0.92	05.02.2013	Moritz Horsch	Updated changes from AG
0.93	22.02.2013	Moritz Horsch	Updated changes from USTUTT
1.0	28.02.2013	Moritz Horsch	Final version

Table of Contents

1. Introduction

1.1 Scope

The scope of this document is to provide a examination of the various standards for interface devices on computer and mobile platforms in order to provide a well-engineered definition of the requirements for the IFD service of the FutureID client.

1.2 Outline

This document is structured as follows: Section 2 provides an overview of existing standards for interface devices which are considered in the requirement analysis. Section 3 describes the requirements for the IFD service. Section 4 summarizes the results and gives a brief overview of the intended architecture and functionality.

1.3 Terminology

1.3.1 Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [10].

1.3.2 Abbreviations and Notations

APDU	Application Protocol Data Unit
API	Application Programming Interface
CA	Certificate Authority
ICC	Integrated Circuit Card
IFD	Interface Device
MTM	Mobile Trusted Module
NFC	Near Field Communication
PC/SC	Personal Computer/Smart Card
SCIO	Smart Card Interface Input Output
SICCT	Secure Interoperable Chip Card Terminal
TPM	Trusted Platform Module

1.3.3 Terms

Secure Element	A secure element is a tamper-resistant device and capable of securely storing data and hosting applications.

2. Existing Interface Device Standards

This section provides an overview of existing standards for interface devices on computer and mobile platforms. Based on this overview, the requirements for the IFD service of the FutureID client are defined.

We consider NFC, Open Mobile API, PC/SC, SICCT, and TPM. The NFC technology provides communication to contactless smart cards and is suitable for mobile devices. The Open Mobile API enables applications to access and support different Secure Elements in mobile devices. PC/SC is a widespread interface to access smart cards via a common interface. SICCT provides an interface to access smart cards and is particularly used in the German e-Health infrastructure. TPM is a secure and trustworthy hardware module for computer platforms.

2.1 NFC

Near Field Communication (NFC) [11] [12] [2] [3] is a wireless communication technology. A connection is established by bringing devices into close proximity. NFC can be used for data exchange, contactless transactions, and the configuration of more complex communication protocols. The nominal transmission range is only a few centimeters and the data rate is up to 424 kbit/s. NFC is compatible to ISO/IEC 14443 [13] and therefore allows communicating with passive tags and contactless smart cards.

2.2 Open Mobile API

The Open Mobile API as specified by the SIMalliance [5] is an API that allows applications on mobile devices to access various kinds of secure elements in a standardized way. These elements can be SIM cards, secure microSD cards or other kinds of secure tokens embedded in or attached to the mobile device. The API definition is independent of a specific platform or programming language and could therefore be implemented on any type of device and operating system. A general overview of the architecture is shown in Figure 1.

The Transport Layer provides general access to the secure elements via Application Protocol Data Units (APDU) and acts as the foundation of the Service Layers. Communication takes place based on the concept of the ISO/IEC 7816-4 [14] standard with logical channels separating communication between different applications.

As the next higher layer, the Service Layer provides a higher abstraction of different secure element functions. Therefore, they are easier to use by application developers than the underlying Transport API. The Service Layer relies on the transport layer for communicating to the secure element and consists of different APIs for specialized purposes, like file management and authentication. For the integration of future new types of elements, a SE provider interface can be added. The highest layer, the Application Layer, represents the various applications on the mobile device that make use of the SE functionality, e.g., a payment application, a signature application or an eID.

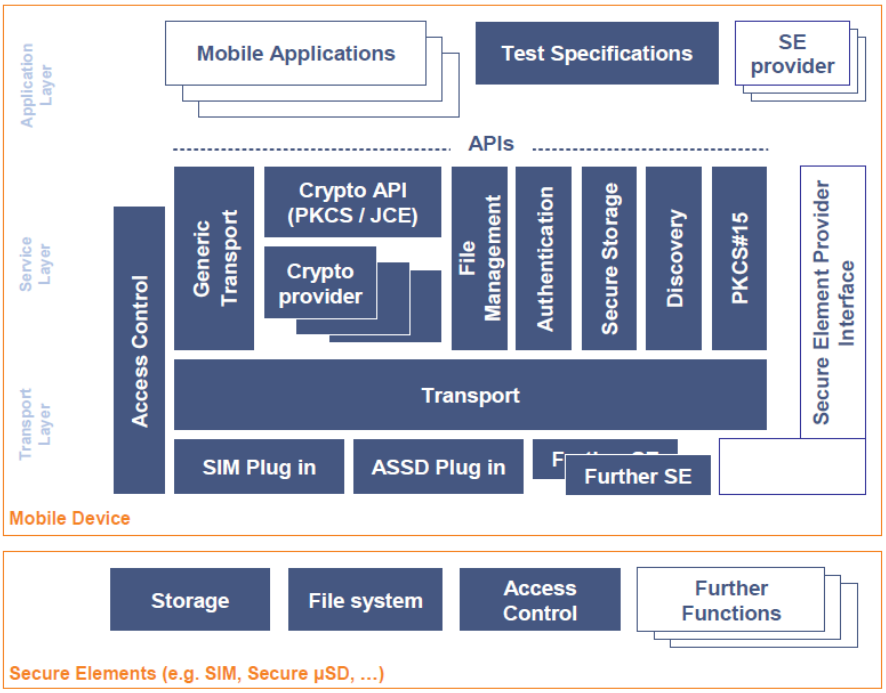

Figure 1: General Architecture of the Open Mobile API (from [5]).

2.3 PC/SC

The Personal Computer/Smart Card (PC/SC) Workgroup Specifications [1] define a common interface to access Integrated Circuit Cards (ICC) from within different computing environments. PC/SC can be used to access ICCs with contacts as defined in ISO/IEC 7816 [15] as well as contactless ICCs as defined in ISO/IEC 14443 [13]. It is an integral part of Windows since Windows 2000 [16]. An open source implementation of PC/SC is available from the PC/SC Lite Project [17] for Linux and Unix. A forked version of PC/SC Lite is distributed with Mac OS X [18].

2.4 SICCT

The Secure Interoperable Chip Card Terminal (SICCT) specification [4] defines a generic concept for application-independent card terminals which is based on established card terminal and ICC standards including ISO/IEC 7816 [15], ISO/IEC 14443 [13] and PC/SC [1]. The objective of the SICCT specification is to harmonize the different standards with respect to security and interoperability and to ease the integration, the configuration and the secure operation of card terminals. SICCT defines a high-level API to access card terminals and smart card applications. It is particularly used in the German e-Health sector.

2.5 TPM and MTM

The Trusted Platform Module (TPM) [6] is a piece of hardware (chip) that can perform certain trust and security related tasks, as specified by the Trusted Computing Group (TCG). This device is typically bound to the platform and cannot be removed. The nature of the hardware implementation provides for better protection against attacks, such as man-in-the-middle or private key extraction. The TPM cannot actively initiate operations or communications with other devices. The main capabilities are secure generation of cryptographic keys, secure storage, ensuring confidentiality of stored data and authentication of the platform, as well as binding data to a platform and ensuring the integrity of the platform by various measurements. The specification of version 1.2 has been published as ISO/IEC 11889 [19].

Every TPM has an Endorsement Key (EK) which uniquely identifies it. This key should be backed by an EK certificate which is issued by the TPM manufacturer. If this certificate is present, it is possible to identify the module as a genuine device. Using a trusted third party, the so-called Privacy CA, one can prove the existence and state of a genuine TPM and in general also the state of the system as such to a remote entity (Remote Attestation). To circumvent the inherent privacy issues caused by the uniqueness of the EK, so-called Attestation Identity Keys (AIKs) can be used instead. These keys are aliases for the EK and cannot be traced back directly to the specific module. It might be inevitable to ensure not only the genuineness of the TPM, but also that a private key for a corresponding public key certificate was generated and stored inside the TPM. Therefore, the TCG specified the Subject Key Attestation Evidence (SKAE) X.509 certificate extension [20]. This extension provides a mechanism to join TCG-oriented security assertions within a common certificate.

The Mobile Trusted Module (MTM) [21] is the equivalent of a TPM on a mobile platform and was specified by the TCG. Similar to the TPM, the MTM provides trusted resources via a subset of the standard TPM 1.2 structures and commands. The main difference to the TPM is that MTMs can be implemented in hardware and/or software. In addition, the MTM has some mobile device specific features, e.g., the Reference Integrity Metric (RIM) certificates. However, it seems that currently no mobile device exists that implements the MTM standard and this will most likely not change in the foreseeable future. Hence, we will not consider MTM in the requirements of the IFD service.

3. Requirements

The following section provides the requirements for the IFD service.

3.1 Generalized Interface

3.1.1 Smart Cards and Card Terminals

The IFD service MUST provide a generalized interface for communication with arbitrary card terminals and smart cards. It SHOULD contain functions for card terminals like establish and destroy a session, status and capability information and control commands. In addition, the IFD SHOULD contain commands to connect, disconnect and transmit data to a smart card.

3.1.2 Secure Elements

The IFD service SHOULD provide a generalized interface for communication with arbitrary secure elements.

3.2 Protocols

The IFD service MUST support protocols to establish trusted channels. It SHOULD provide commands to establish and destroy trusted channels. Input data, which is required to establish the channel, SHOULD be defined in a flexible and transparent way. The IFD SHOULD consider protocol specific requirements like the functionality of connected card terminals, for instance, whether a keypad or display is present or not.

3.3 Support of Existing Standards

The IFD service considers the following existing standards.

3.3.1 NFC

The IFD service MUST support Near Field Communication (NFC).

3.3.2 Open Mobile API

The IFD service SHOULD support the Open Mobile API.

3.3.3 PC/SC

The IFD service MUST support PC/SC.

3.3.4 SICCT

The IFD service SHOULD support SICCT.

3.3.5 TPM

The IFD service SHOULD support Trusted Platform Modules (TPM).

3.4 Proxy

Mobile and computer-based platforms MAY support multiple interfaces. Consider a notebook that is equipped with an NFC and a USB interface. Thus, smart cards can be accessed via NFC or via a USB-connected card reader. Therefore, the IFD service SHOULD support multiple interfaces mentioned in Section 3.3 simultaneously.

3.5 ISO/IEC 24727-4

The IFD service MUST support the ISO/IEC 24727-4 [7] specification.

3.6 Bindings

The IFD service MUST provide a common interface for external application. The interface SHOULD support different bindings.

3.6.1 Common language-specific Programming Interface

The IFD service SHOULD support a language-specific Programming Interface, for instance for Java and C.

3.6.2 SOAP

The IFD service SHOULD support a SOAP [8] binding.

3.6.3 PAOS

The IFD service SHOULD support a PAOS [9] binding.

4. Conclusion

The IFD service will provide a interface for communication to various devices. It will provide a generalized interface for smart cards and card terminals and will support existing standards like NFC [2] [3], Open Mobile API [5], PC/SC [1], and SICCT [4]. The IFD will be based on ISO/IEC 24727-4 [7] and will integrate different bindings like SOAP [8] and PAOS [9].

The ISO/IEC 24727-4 standard offers a well-established foundation for the IFD service for the FutureID client. The integration of existing standards like PC/SC and Open Mobile API provides support for various devices like smart cards and secure elements. The IFD will support computer and mobile environments and therefore a wide variety of different platforms.

Section 4.1 provides an overview of the intended architecture and Section 4.2 outlines the intended functionality of the IFD service.

4.1 **Architecture**

The intended architecture of the IFD service is depicted in Figure 2. At the top the IFD service provides a common and homogeneous interface for a standardized usage of different device interfaces. The IFD includes an interface to integrate protocols through which applications can establish a trusted channel to smart cards and secure elements. In addition, the IFD service provides a common interface for Smart Card Input/Output (SCIO) and card terminal operations. At the bottom of the architecture the device interfaces of the certain technology are integrated.

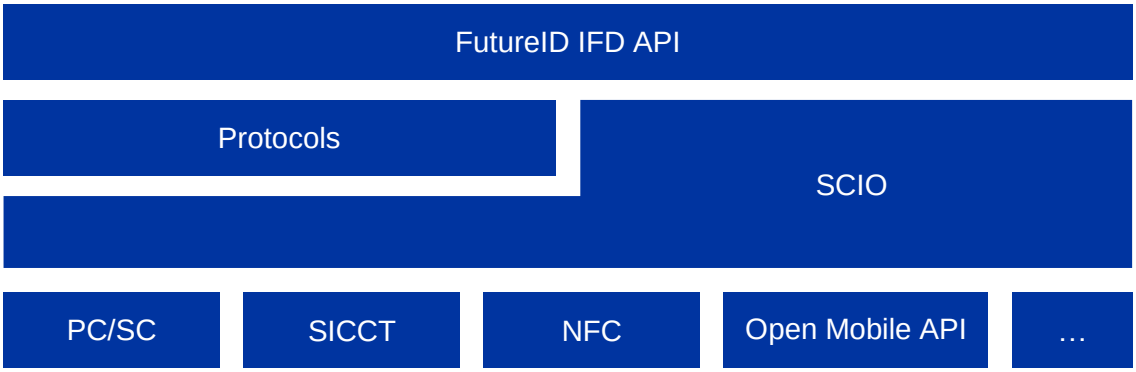

Figure 2: IFD architecture for smart cards

The architecture depicted in Figure 2 focuses on ISO/IEC 7816 based smart cards and card terminals using Application Protocol Data Units (APDU). However, the TPM interface uses different data structures. Hence, TPMs will not be integrated as an additional interface at the bottom of the architecture as depicted in Figure 2.

To facilitate the usage of various device interfaces simultaneously, the IFD service should provide an additional layer which acts as a proxy (cf. Figure 3). The IFD proxy allows applications to simultaneously access devices, which are connected through different interfaces, in a transparent manner.

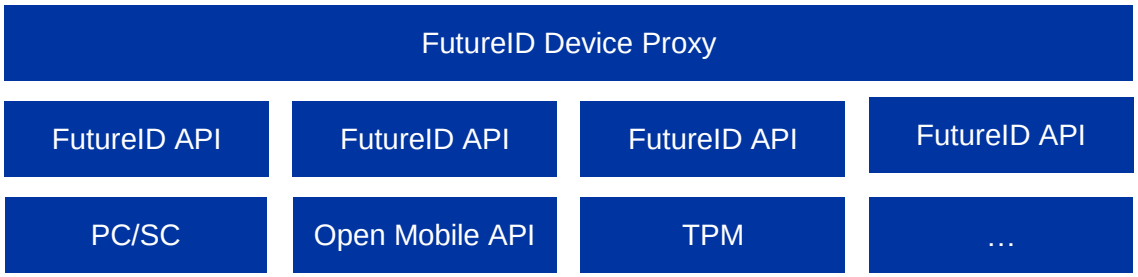

Figure 3: IFD Proxy architecture

4.2 Functionality

This section gives a short overview of the intended IFD functions. A detail description will be provided in the forthcoming deliverable D31.2.

The following tables contain the respective functions, a reference to the requirement mentioned in Section 3 and their levels.

4.2.1 Slot Device related

The slot device related functions consider the capacities of the slot device.

	Function	Requirement	Level
F1	Get capability information	3.1.1	SHOULD
F2	Establish and destroy session	3.1.1	MUST
F3	Common IFD control commands	3.1	SHOULD

4.2.2 Slot related

The slot related functions focus on the capacities of the device attached in the slot.

	Function	Requirement	Level
F4	Connect and disconnect to a device in a particular slot of an IFD	3.1.1, 3.1.2	MUST
F5	Establish a trusted channel	3.2	MUST
F7	Start and end transactions	3.1.1, 3.1.2	MUST
F8	Transmit data	3.1.1, 3.1.2	MUST

4.2.3 User related

The user related functions consider the user interaction of the IFD service.

	Function	Requirement	Level
F9	Verify and modify credentials	3.2	MUST

5. Bibliography

[1] PC/SC Workgroup, *PC/SC Workgroup Specifications,* Version 2.01.11, Part 1 - 10, 2012.

[2] ISO/IEC, *Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol (NFCIP-1),* International Standard, ISO/IEC 18092, 2003.

[3] ISO/IEC, *Information technology - Telecommunications and information exchange between systems - Near Field Communication Interface and Protocol -2 (NFCIP-2),* International Standard, ISO/IEC 21481, 2005.

[4] TeleTrusT Deutschland e.V., *SICCT - Secure Interoperable ChipCard Terminal,* Version 1.6, 2009.

[5] SIMalliance, *Open Mobile API specification,* Version 2.03, 2012.

[6] Trusted Computing Group, *TPM Main,* Version 1.2, Part 1 - 3, 2011.

[7] ISO/IEC, *Identification cards - Integrated circuit card programming interfaces - Part 4: Application programming interface (API) administration,* International Standard, ISO/IEC 24727-4, 2008.

[8] B. Don, E. David, K. Gopal, L. Andrew and M. Noah, Simple Object Access Protocol (SOAP) 1.1, 2000.

[9] A. Robert and K. John, Liberty Reverse HTTP Binding for SOAP Specification, Liberty Alliance Specification, Version 2.0, 2006.

[10] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels,* RFC 2119, 1997.

[11] ECMA International, *Standard ECMA-340 - Near Field Communication Interface and Protocol (NFCIP-1),* 2004.

[12] ECMA International, *Standard ECMA-352 - Near Field Communication Interface and Protocol -2 (NFCIP-2),* 2003.

[13] ISO/IEC, *Identification cards - Contactless integrated circuit cards - Proximity cards,* International Standard, ISO/IEC 14443, Part 1 - 4, 2008 - 2011.

[14] ISO/IEC, *Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange,* International Standard, ISO/IEC 7816, Part 4, 2005.

[15] ISO/IEC, *Identification cards - Integrated circuit cards,* International Standard, ISO/IEC 7816, Part 1 - 15, 1999 - 2011.

[16] Microsoft Corporation, “Smart Cards,” [Online]. Available: http://technet.microsoft.com/en-us/library/bb742533.aspx. [Accessed 07 12 2012].

[17] D. Corcoran and L. Rousseau, “PCSC lite project,” [Online]. Available: http://pcsclite.alioth.debian.org/pcsclite.html. [Accessed 07 12 2012].

[18] Mac OS Forge, “SmartCard Services,” [Online]. Available: https://smartcardservices.macosforge.org/. [Accessed 07 12 2012].

[19] ISO/IEC, *Information technology - Trusted Platform Module,* International Standard, ISO/IEC 11889, Part 1 - 4, 2009.

[20] Trusted Computing Group Infrastructure Workgroup, *Subject Key Attestation Evidence Extension,* Version 1.0, 2005.

[21] Trusted Computing Group, *TCG Mobile Trusted Module Specification,* Version 1.0, 2010.