From the eCard-API-Framework towards a comprehensive eID-framework for Europe

Dr. Detlef Hühnlein[1] · Manuel Bach[2]

[1]secunet Security Networks AG,
Sudetenstraße 16, 96247 Michelau, Germany
detlef.huehnlein@secunet.com

[2] Federal Office for Information Security (BSI)
Godesberger Allee 185-189, D-53175 Bonn, Germany
manuel.bach@bsi.bund.de

Abstract

The German eCard-strategy aims at harmonizing the various government projects which issue or use smart cards for authentication and signature purposes. Against this background the German government developed the eCard-API-Framework **[eCard-TR]** which aims at supporting arbitrary smart cards and facilitating the integration of them into various eID-applications. The present contribution provides a brief overview of the main features of the eCard-API-Framework, highlights current standardization efforts at CEN and ISO and provides an outlook how this approach might form the basis for a comprehensive eID-framework for Europe and beyond.

1 Introduction

The German eCard-strategy (cf. **[Kowa07]**) aims at harmonizing the various government projects, which issue or use smart cards for authentication and signature purposes. Within these projects there will be issued around 80 Million electronic health cards (eHC) **[eHC]** and in a second step about the same number of electronic ID (eID) cards which will comply to the forthcoming specification of the European Citizen Card standardized in **[prCEN15480-1]** and **[prCEN15480-2]**. These cards as well as all privately issued banking and signature cards shall smoothly interoperate with major public applications, like the German eTax-application (ELektronische STeuerERklärung, ELSTER) and various other eGovernment initiatives. As this comprises a very heterogeneous set of smart cards, this plan induces a major challenge for the related smart card middleware.

Against this background the Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik, BSI) developed a set of platform-independent interfaces - called the eCard-API-Framework **[eCard-TR]** - which aims at supporting arbitrary smart cards and facilitating the integration of them into existing and forthcoming eID-applications.

As there are similar requirements for handling electronic identities in other European countries, it is natural to think about applying the eCard-API-Framework for cross-border processes and in other countries as well.

The current paper provides a brief overview of the eCard-API-Framework (Section 2) and sketches possible path towards developing a comprehensive eID-framework for Europe and beyond (Section 3).

2 The eCard-API-Framework

This section provides a brief overview of the eCard-API-Framework as specified in **[eCard-TR]** and is structured as follows: Section 2.1 highlights the main features of the eCard-API-Framework. Section 2.2 provides an overview of the architecture of the framework. At the heart of the eCard-API-Framework are the Generic Card Services, which are addressed in Section 2.3. Finally Section 2.4 will provide an overview of possible deployment scenarios.

2.1 Main features

The main features of the eCard-API-Framework may be summarized as follows.

2.1.1 Truly generic card services

The eCard-API-Framework is build around a web-service based variant of the forthcoming **[ISO24727-3]**-interface. However instead of using the underlying so called "generic card interface" defined in **[ISO24727-2]**, which in fact is – considering the range of existing and evolving European eID projects – neither generic nor sufficient, the eCard-API-Framework provides a generalization of both **[ISO24727-2]** and **[ISO7816-15]** which works with arbitrary smart cards, as the card-specific information is provided to the middleware in form of XML-based `CardInfo` files (cf. Section 2.3). These files allow to recognize the type of the card and map the generic **[ISO24727-3]**-calls to card-specific APDUs, even if – which is the case for most issued eID-cards – the card is not fully compliant to **[ISO7816-15]**.

2.1.2 Platform-independent and scalable

As the eCard-API-Framework is a collection of web-service-interfaces, it is equally easy to support this interface in both Java- or .NET-based applications. Due to the web-service-interfaces it is furthermore trivially possible to scale from a local deployment on a single "client" machine to a fully distributed networked "server" environment. Hence it is possible to support not only the "Loyal-Stack" (**[ISO24727-4]**, Section 5.2) but also the "Remote-Loyal-Stack" (**[ISO24727-4]**, Section 5.4) and the "Remote-ICC-Stack" (**[ISO24727-4]**, Section 5.6) using the same set of interfaces. Please refer to Section 2.4 for a more comprehensive discussion of possible deployment scenarios.

2.1.3 Support for and abstraction of various card terminal technologies

The eCard-API-Framework supports a wide range of existing and emerging card terminal technologies. Therefore it is possible to use local **[PC/SC]**-IFDs, networked **[SICCT-v1.1]**-terminals as well as proprietary RFID/MRTD-readers over the same interface. The Reader-Interface from a previous version of the eCard-API-Framework has recently been harmonized with the Card Transport Interface from the **[Onom@topic]**- project and has been included as "Interface-Device API" in **[prCEN15480-3]** and was recently submitted as contribution for **[ISO24727-4]**.

2.1.4 Developer-friendly “high-level”-interfaces

While it is possible to use the very fine granular “low-level”-calls (e.g. for card services and other cryptographic functions) it is also possible to create and verify complex advanced electronic signatures in popular standard formats (e.g. XAdES or CMS) involving complex PKI-services with very simple API-calls. In a similar manner it is possible to perform complex card-to-card-authentication protocols with a single `Card2CardAuthenticate`-call with two parameters (`FromCard` and `ToCard`).

2.1.5 Aligned with major eID-standards and Microsoft’s cryptographic architecture

In the design of the framework it was a major goal to align it with important eID- and related PKI-standards, such as **[CEN14171]**, **[CEN14890-2]**, **[prCEN15480-1]**, **[prCEN15480-2]**, **[prCEN15480-3]**, **[ETSI-101733]**, **[ETSI-101903]**, **[ETSI-102231]**, **[ISO24727-1]**, **[ISO24727-2]**, **[ISO24727-3]**, **[ISO24727-4]**, **[OASIS-DSS]**, **[PC/SC]**, **[RFC2560]**, **[RFC3161]**, **[RFC3369]**, **[SICCT-v1.1]** and **[XML-Enc]**. Furthermore the framework was designed such that there are no negative interferences with Microsoft’s cryptographic architecture **[MS-CAPI]** and the generic card services block neatly fits into the forthcoming Next Generation Cryptographic API **[MS-CNG]** and may be used to provide a truly generic “key storage provider”.

2.1.6 May support higher Identity Management Layers

The eCard-API-Framework may form the basis for (federated) identity management solutions, like **[CardSpace]**, **[Higgins]** and/or **[Liberty]**. As the eCard-API-Framework as well as all these approaches for (federated) identity management are built upon widely recognized web-service standards (WSDL etc.), it is possible to build a comprehensive eID-framework by combining the different layers (cf. Section 3).

2.2 Architecture

As depicted in Fig. 1 the eCard-API-Framework roughly consists of the following layers, whose functionality is briefly described in the following:

- Application-Layer
- Identity-Layer
- Authentication-Layer
- Terminal-Layer

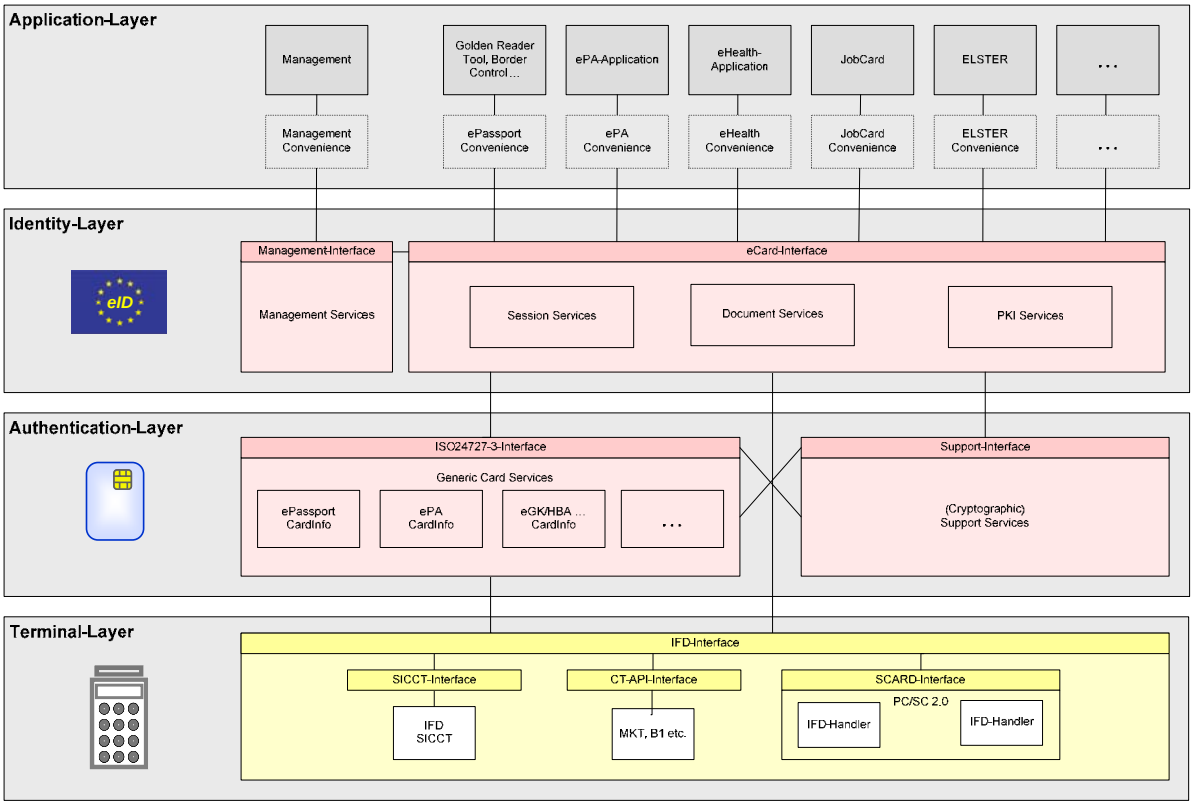


Fig. 1: eCard-API-Framework

2.2.1 Application-Layer

The Application-Layer may comprise different applications which access the services provided by the eCard-API-Framework in order to access eID-functions, secure electronic documents by means of (advanced) electronic signatures and/or encryption or to obtain access to some electronic service provided by a Service Provider.

2.2.2 Identity-Layer

The Identity-Layer provides functions to establish secure sessions (e.g. using **[RFC2246]**) and secure documents in various formats by means of (advanced) electronic signatures (e.g. according to **[ETSI-101733]** or **[ETSI-101903]**) and encryption (e.g. using **[RFC3369]** or **[XML-Enc]**). The functions for generating and verifying electronic signatures are closely aligned with the recently finalized standard **[OASIS-DSS]**.

2.2.3 Authentication-Layer

The Authentication-Layer provides the basic authentication functionality using arbitrary smart cards. The authentication services are accessed using the “Service Access Interface” which is currently standardized in **[ISO24727-3]** and **[prCEN15480-3]**. In order to be able to use arbitrary identity tokens – especially tokens which fail to provide a standardized cryptographic information application according to **[ISO7816-15]** – the generic card services will use the XML-based `CardInfo`-structure introduced in **[HüBa07]** and currently discussed in CEN TC 224 WG 15 (cf. Section 2.3).

2.2.4 Terminal-Layer

The Terminal-Layer provides a homogeneous interface for arbitrary card terminals, which is currently standardized in **[prCEN15480-3]** and **[ISO24727-4]**. While the basic functions of this interface are similar to **[PC/SC]** this interface also allows to use more sophisticated interface devices according to **[SICCT-v1.1]** for example, which support multiple slots (for contact-based and/or contactless cards) as well as functional units (e.g. display, keypad, biometric sensors).

2.3 Generic Card Services

At the heart of the eCard-API-Framework is the generic card services block underneath the ISO24727-3-interface. This block is able to handle arbitrary eID-cards, since the necessary information to recognize some card type and to map the generic ISO24727-3-calls to card-specific APDUs is provided in an XML-based `CardInfo`-file, as outlined in Fig. 2.

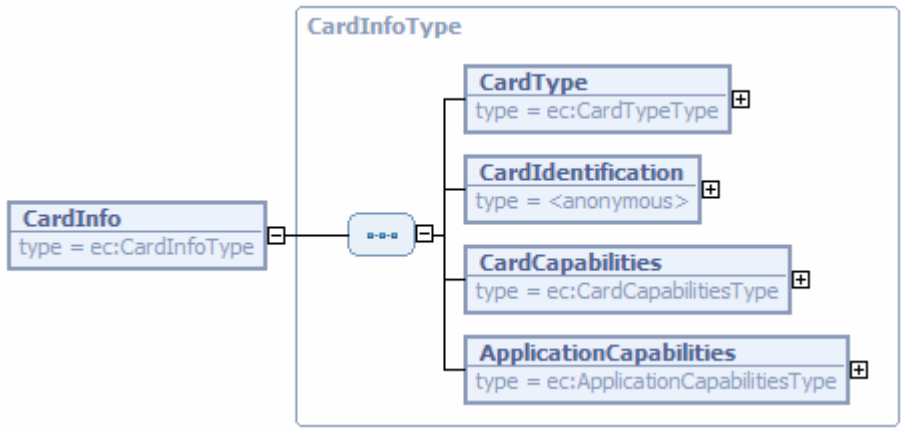


Fig. 2: `CardInfo`-Structure

The `CardInfo`-files roughly comprises of the following elements:

- `CardType` provides a unique identifier for the present (version of the) card type.
- `CardIdentification` comprises the necessary information to recognize the card type by a set of characteristic features (e.g. contents of the ATR/ATS and specific files on the card). As explained in Section 2.2 of **[HüBa07]** these data may be used to build a "decision tree" which is used to recognize the card type.
- `CardCapabilities` provide information concerning the capabilities of the specific card (e.g. basic card capabilities according to **[ISO7816-4]** as well as the cryptographic and biometric capabilities of the card).
- `ApplicationCapabilities` contain information concerning the personalization of the eID-card with card applications, keys (called Differential-Identities in **[ISO24727-3]**) and other data stored on the card (Data Structures for Interoperability (DSI) grouped in Data Sets according to **[ISO24727-3]**).

More details concerning these structures and the use of this information for mapping the generic **[ISO24727-3]**-calls to card-specific APDUs are provided in Section 2.3 of **[HüBa07]** and **[eCard-TR]**.

2.4 Deployment Scenarios

In this section we briefly sketch some possible deployment scenarios of the eCard-API-Framework to demonstrate the flexibility of this approach.

2.4.1 Loyal-Stack

The simplest deployment scenario is the so called "Loyal-Stack" (cf. **[ISO24727-4]**, Section 5.2) in which all layers of the eCard-API-Framework as depicted in Fig. 1 are deployed on a single machine. Besides the simplest case in which there is only one smart card accessed, the eCard-API-Framework is also able to handle more sophisticated Loyal-Stack scenarios. For example it is possible to support use cases for "Card-to-Card-Authentication" (cf. Fig. 3) or "Comfort Signature" (cf. Fig. 4), which are both (about to be) used in German eHealth-applications.

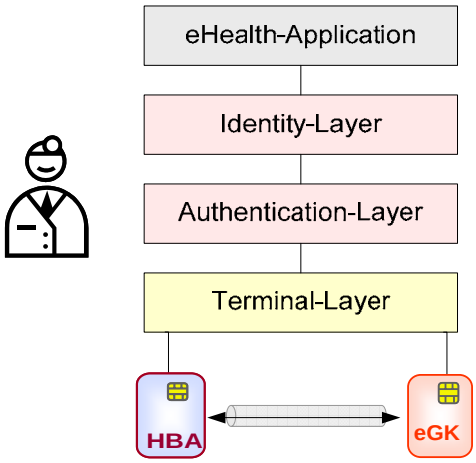


Fig. 3: Card-to-Card-Authentication

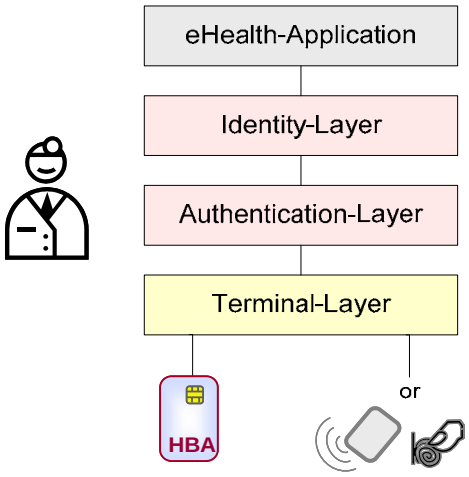


Fig. 4: Comfort Signature

For the first scenario it is possible to start an authentication between two cards using the complex protocols defined in **[CEN14890-2]** using a simple API call (`Card2CardAuthenticate`), which only specifies which keys on the cards are to be used for this authentication step.

For the sake of usability of qualified electronic signatures for power users (e.g. doctors, which issue hundreds of electronic prescriptions per day) it was proposed to activate the secure signature creation device once a day with the required six character signature PIN and perform the wilful act to sign a specific document with a more comfortable authentication mechanism in which a finger print is captured or an RFID-token is presented. Such deployment scenarios are easily supported by the eCard-API-Framework, because it is built around **[ISO24727-3]**, which uses (an arbitrary Boolean combination of) Differential-Identities (DIDs) to protect resources.

2.4.2 Remote-Loyal-Stack

The "Remote-Loyal-Stack" (cf. **[ISO24727-4]**, Section 5.4) is somewhat more complicated as the different layers are deployed on different machines. While the Identity-Layer (cf. Fig. 1) is present on both machines, the other layers may or may not be present depending on the specific application. In the following we will briefly highlight some popular examples for the "Remote-Loyal-Stack".

Similar to the local card-to-card-authentication (cf. Fig. 3) it is possible to perform the protocols defined in **[CEN14890-2]** over the internet. This may be used to handle electronic prescriptions in an internet-pharmacy (cf. Fig. 5) for example.

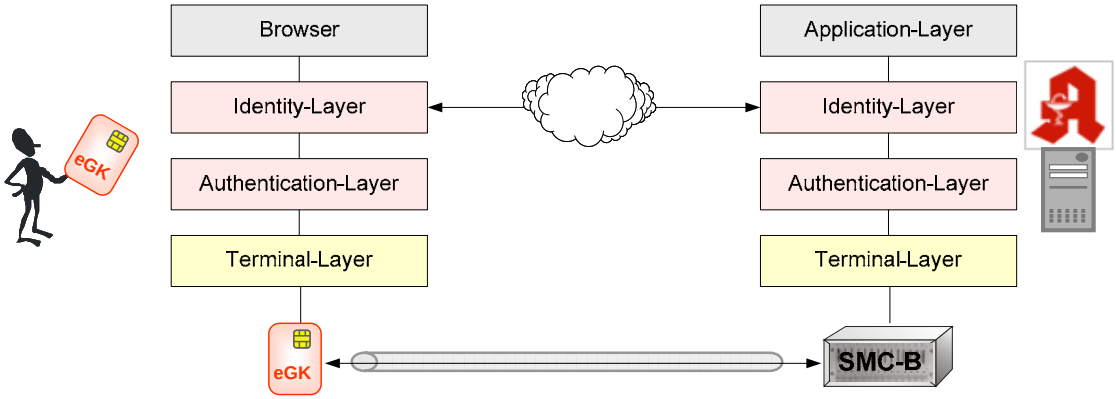
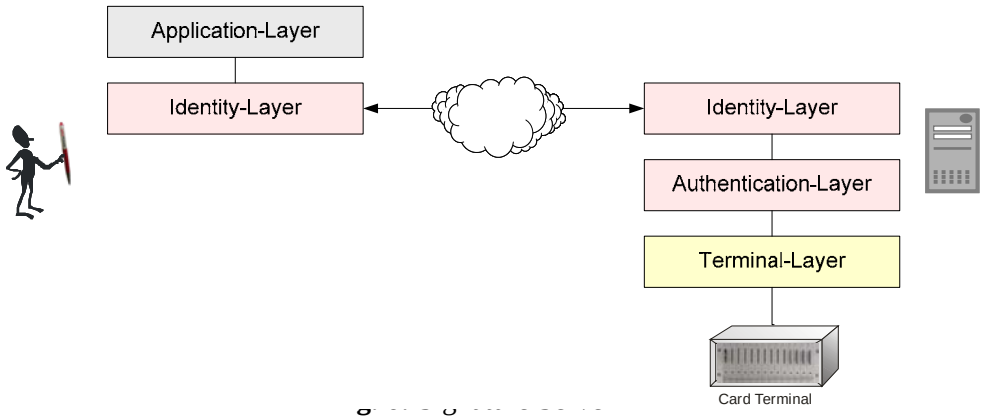


Fig. 5: Internet-Pharmacy

If an application (cf. Fig. 6) wants to sign invoices it does not necessarily need the Authentication- and Terminal-Layer on the local machine but it may use its Identity-Layer to connect to a Signature Server in order to get the invoices signed.



On the other hand it would be possible to implement a “Remote-Loyal-Stack” scenario for web signing purposes (cf. Fig. 7) in which the application logic is present at the server but the signature generation uses the Authentication- and Terminal-Layer of the user’s local machine.

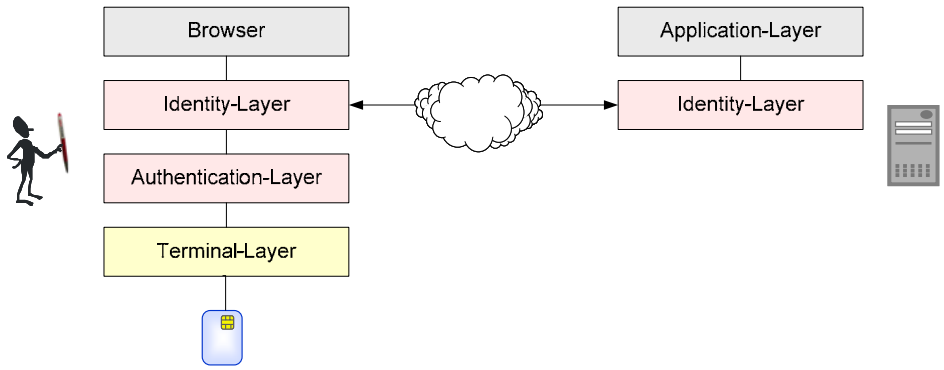


Fig. 7: Web-Signing

While it is not part of the current specification **[eCard-TR]** yet it would also be possible to support mobile signatures according to **[ETSI-102204]** (cf. Fig. 8) as an additional "Remote-Loyal-Stack" scenario with only moderate extensions.

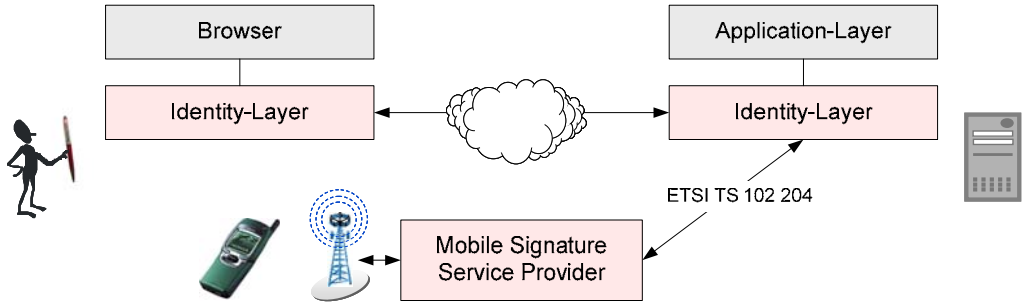


Fig. 8: Mobile Signature

2.4.3 Remote-ICC-Stack

Finally it is also possible to have the Authentication- and Terminal-Layer running on different machines in order to obtain the "Remote-ICC-Stack" (cf. **[ISO24727-4]**, Section 5.6). As depicted in Fig. 8 this may be used for Card-Application Management Services (CAMS) as existing for the German eHealth-card **[eHC]**.

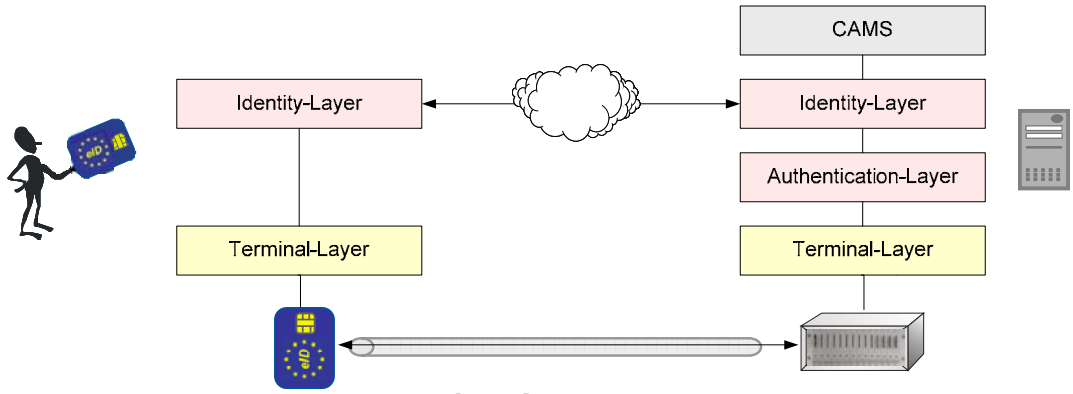


Fig. 8: Card-Application-Management

3 Towards a comprehensive eID-framework for Europe

With the Manchester Ministerial Declaration **[ManDec05]**, the importance for interoperable identity management for our Information Society has been formally put on the political agenda of Europe and recognized as a priority and key enabling factor for our development. The declaration confirms the national autonomy in the issuance of national identity documents and electronic identities. Hence there may be various eIDs using different processes and technologies and hence providing different levels of trust. As the eIDs themselves may significantly differ from one domain to another, it is necessary to provide powerful middleware solutions, which are able to handle arbitrary eIDs and nevertheless provide common interfaces to applications which want to use eID services.

As the eCard-API-Framework **[eCard-TR]** is designed to support arbitrary smart cards (according to ISO/IEC 7816 and ISO/IEC 14443) and has been built on existing and emerging standards (cf. Section 2.1.5) it may well serve as starting point for the development of a com-

prehensive eID-framework for Europe. This development may be part of WP 5 of the forthcoming large scale eID-pilot "STORK" **[Leym07]**. While the eCard-API-Framework may obviously serve as reference architecture for the "Middleware Approach" in WP 5.2 it may also support the "Proxy Approach" in WP 5.3.

To use the eCard-API-Framework as basis for a common European eID-middleware it is only necessary to create appropriate `CardInfo`-files (cf. Fig. 2) for the smart card based eIDs in Europe (and provide appropriate drivers for non-smart-card-based eIDs if necessary).

In order to support the "Proxy Approach" the eCard-API-Framework may serve as basic web service framework for the secure exchange of data in cross border business processes. In this case the interoperability will entirely be handled by the Identity-Layer or an appropriate "Convenience Layer" on top of it. In order to support the different Identity-Management protocols used across Europe it may be necessary to provide different PlugIns for the generic `TC_API_Open`-function specified in **[eCard-TR]** and **[ISO24727-4]**.

4 Conclusion

This paper provided a brief overview of the main features of the eCard-API-Framework **[eCard-TR]** and sketched some possible deployment scenarios. As this approach seems to be among the most comprehensive and flexible approaches for handling smart card based eIDs available today and is closely aligned with existing and emerging international standards we are confident that this approach will sooner or later provide a significant contribution to European eID-interoperability.

References

[CardSpace] Microsoft: *CardSpace*, via http://cardspace.netfx3.com/

[CEN14171] Comité européen de normalisation (CEN): *General guidelines for electronic signature verification*, CEN Workshop Agreement, May 2004, via ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14171-00-2004-May.pdf

[CEN14890-2] Comité européen de normalisation (CEN): *Application Interface for smart cards used as Secure Signature Creation Devices - Part 2: Additional Services*, CEN Workshop Agreement, May 2004, via ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14890-02-2004-May.pdf

[prCEN15480-1] Comité européen de normalisation (CEN): *Identification card systems — European Citizen Card — Part 1: Physical, electrical and transport protocol characteristics*, prCEN/TS 15480-1, proposed Technical Standard, 2007

[prCEN15480-2] Comité européen de normalisation (CEN): *Identification card systems — European Citizen Card — Part 2: Logical data structures and card services*, prCEN/TS 15480-2, proposed Technical Standard, 2007

[prCEN15480-3] Comité européen de normalisation (CEN): *Identification card systems — European Citizen Card — Part 3: European Citizen Card Interope-*

rability using an application interface, prCEN/TS 15480-3, Working Draft, 2007

[eCard-TR] Federal Office for Information Security: *eCard-API-Framework – Technical Directive BSI 03112*, 2007

[eHC] gematik: *The Specification of the German Electronic Health Card eHC, Part 1: Commands, Algorithms and Functions of the COS Platform*, via http://www.gematik.de/upload/gematik_eGK_Specification_Part1_e_V1_1_0_518.pdf
Part 2: Applications and application-related Structures, via http://www.gematik.de/upload/gematik_eGK_Specification_Part2_e_V1_1_1_516.pdf
Part 3: Layout and Physical Properties, vai http://www.gematik.de/upload/gematik_eGK_Specification_Part3_e_V1_1_0_514.pdf

[ETSI-101733] ETSI: *Electronic Signature Formats, Electronic Signatures and Infrastructures (ESI) – Technical Specification*, ETSI TS 101 733 V1.5.1, 2003-12, via http://portal.etsi.org/docbox/EC_Files/EC_Files/ts_101733v010501p.pdf

[ETSI-101903] ETSI: *XML Advanced Electronic Signatures (XAdES)*, Technical Specification, TS 101 903 V1.2.2 (2004-04), via http://uri.etsi.org/01903/v1.2.2/ts_101903v010202p.pdf

[ETSI-102204] ETSI: *Mobile Signature Service - Web Service Interface*, Technical Specification TS 102 204 V1.1.4, via http://portal.etsi.org/docbox/EC_Files/EC_Files/ts_102204v010104p.pdf

[ETSI-102231] ETSI: *Provision of harmonized Trust Service Provider (TSP) status information*, Technical Specification TS 102 231, via http://portal.etsi.org/stfs/STF_HomePages/STF290/draft_ts_102231v010201p&RGW.doc

[Higgins] Higgins Team: *Higgins Trust Framework Project Home*, via http://www.eclipse.org/higgins

[HüBa07] D. Hühnlein, M. Bach: *How to use ISO/IEC 24727-3 with arbitrary Smart Cards*, in C. Lambrinoudakis, G. Pernul, A.M. Tjoa (Eds.): TrustBus 2007, LNCS 4657, SS. 280–289, 2007

[ISO7816-4] ISO/IEC: *Identification cards — Integrated circuit cards — Part 4:Organization, security and commands for interchange*, ISO7816-4, Version 2005-01-15

[ISO7816-15] ISO/IEC: Information technology — Identification cards - Integrated circuit(s) cards with contacts — Part 15: Cryptographic information application, ISO/IEC 7816-15, 2003

[ISO24727-1] ISO/IEC: Identification Cards — Integrated Circuit Cards Programming In-terfaces — Part 1: Architecture, ISO/IEC 24727-1, Final Draft International Standard, 2006

[ISO24727-2] ISO/IEC: Identification Cards — Integrated Circuit Cards Programming In-terfaces — Part 2: Generic Card Interface, ISO/IEC 24727-3, Committee Draft, 2006

[ISO24727-3] ISO/IEC: Identification Cards — Integrated Circuit Cards Programming Interfaces — Part 3: Application Interface, ISO/IEC 24727-3, Committee Draft, 2006

[ISO24727-4] ISO/IEC: Identification Cards — Integrated Circuit Cards Programming Interface — Part 4: API Administration, ISO/IEC 24727-4, Working Draft, 2007

[Kowa07] B. Kowalski: *A survey of the eCard-Strategy of the German Federal Government*, (in German), Proceedings of BIOSIG 2007, GI Lecture Notes in Informatics, 2007

[Leym07] F. Leyman: *e-ID interoperability large scale pilot – STORK*, Talk at the EEMA-Conference "The European e-Identity Conference", Paris, June 2007

[Liberty] Liberty Alliance Project: *The Liberty Alliance*, via http://www.projectliberty.org/

[ManDec05] *Ministerial declaration*, approved unanimously on 24 November 2005, Manchester, United Kingdom, via http://archive.cabinetoffice.gov.uk/egov2005conference/documents/proceedings/pdf/051124declaration.pdf

[MS-CAPI] Microsoft Inc.: *Cryptography Reference (Microsoft CryptoAPI), Platform SDK: Security*, via http://msdn.microsoft.com/library/en-us/security/security/cryptography_reference.asp

[MS-CNG] Microsoft Inc.: *Cryptography API: Next Generation* http://msdn.microsoft.com/library/en-us/seccrypto/security/cryptography_api__next_generation.asp

[OASIS-DSS] OASIS: *Digital Signature Service Core Protocols, Elements, and Bindings*, Version 1.0, OASIS Standard, via http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf

[Onom@topic] M. Faher: *Onom@Topic – project*, presentation at Porvoo 9, May 2006, via http://porvoo9.gov.si/pdf/FRI_15_1000_MFaher_AXALTO_Porvoo9.pdf

[PC/SC] PC/SC Workgroup: *PC/SC Workgroup Specifications 1.0/2.0*, via http://pcscworkgroup.com

[RFC2246] T. Dierks, C. Allen: *The TLS Protocol - Version 1.0*, RFC2246, via http://www.ietf.org/rfc/rfc2246.txt

[RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: *X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol – OCSP*, IETF RFC 2560, via http://www.ietf.org/rfc/rfc3161.txt

[RFC3161] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*. IETF RFC 3161, August 2001, via http://www.ietf.org/rfc/rfc3161.txt

[RFC3369] R. Housley: *Cryptographic Message Syntax (CMS)*, IETF RFC 3369., via http://www.ietf.org/rfc/rfc3369.txt

[SICCT-v1.1] TeleTrusT: *SICCT-Spezifikation*, Version 1.1.0, 2006-12-19, via http://www.teletrust.de/fileadmin/files/publikationen/Spezifikationen/SICCT_Spezifikation_1.10.pdf

[XML-Enc] W3C Recommendation: *XML Encryption Syntax and Processing*, 10. Dezember 2002, via http://www.w3.org/TR/xmlenc-core/