An Open eCard Plug-in for accessing the German national Personal Health Record

Raik Kuhlisch[1] · Dirk Petrautzki[2] · Johannes Schmölz[3] · Ben Kraufmann[1]
Florian Thiemer[1] · Tobias Wich[3] · Detlef Hühnlein[3] · Thomas Wieland[2]

[1] Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{raik.kuhlisch,ben.kraufmann}fokus.fraunhofer.de

[2] Hochschule Coburg, Friedrich-Streib-Str. 2, 96450 Coburg, Germany
{petrautzki,thomas.wieland}hs-coburg.de

[3] ecsec Gmbh, Sudetenstraße 16, 96247 Michelau, Germany
{johannes.schmoelz,tobias.wich,detlef.huehnlein}@ecsec.de

Abstract: An important future application of the German electronic health card (elektronische Gesundheitskarte, eGK) is the national Personal Health Record (PHR), because it enables a citizen to store and retrieve sensitive medical data in a secure and self-determined manner. As the stored data is encrypted with an eGK-specific certificate and retrieving the encrypted data is only possible after TLS-based authentication, the citizen needs to use a so called "PHR Citizen Client", which allows to use the eGK for strong authentication, authorization, and decryption purposes. Instead of building such an application from scratch, this paper proposes to use the Open eCard App and its extension mechanism for the efficient creating of a PHR Citizen Client by developing an Open eCard Plug-in for accessing the German national Personal Health Record.

1 Introduction

Usually the implementation of smart card based applications is fairly challenging, because it requires a profound knowledge of several technologies: communication with diverse card terminals, reading and writing data on various smart card models and performing cryptographic operations for authentication, signature and encryption. Furthermore, such a smart card based application should remain usable and secure if new types and generations of smart cards and terminals as well as operating systems come into market.

Due to several emerging application domains around e-government and e-health in Germany based on the electronic identity card (neuer Personalausweis, nPA) or the electronic health card (elektronische Gesundheitskarte, eGK), providing smart card based applications is a key objective for many software vendors. Against this background an important question is how software vendors can provide smart card based applications *without* focusing on smart card technology and security infrastructures but mainly on their application-specific expertise.

In 2005 the Federal Government of Germany launched the eCard strategy [HoSt11] – a plan that aims at harmonizing different smart card based governmental IT projects. Based on this preparatory work, the specifications developed by the German Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik) [BSI-TR-03112] and first-hand experiences from projects in the context of the electronic health card, the project ID4health was started. The project group of ID4health develops a comprehensive architecture for federated identity management for the public health sector. Part of this architecture is a set of secure services that allow for using the electronic health card and the electronic health professional card. These services rely on the platform independent open source implementation of the eCard-API framework – the Open eCard App [HPS+12].

In order to demonstrate the feasibility of the ID4health approach, the German national Personal Health Record (PHR) was chosen. In this paper we present our approach on how to securely access electronic health information about individual patients using the electronic health card and the extensible Open eCard platform.

The rest of the paper is structured as follows: In Section 2 we briefly introduce the German national PHR according to the health card act, highlight the key aspects, and give a short overview of its architecture and security aspects. In Section 3 we briefly describe the extensible Open eCard platform including the architecture, the interfaces, and sketch how the extension mechanism works. In Section 4 we present our approach to integrate the Open eCard App with a PHR application for the citizen to fulfill the security requirements of accessing a PHR without implementing the security mechanisms in the PHR-specific desktop or mobile application. In Section 5 we discuss our solution and provide an overview of the next steps.

2 German national Personal Health Record

In 2009 the German Federal Ministry of Health initiated an R&D project on the design and prototypical implementation of a German national PHR. Project participants were the Fraunhofer Society, the Federation of German Clinical Research Networks (TMF), and the University Medical Center Göttingen (UMG). Associated contractors have been the German Hospital Federation (Deutsche Krankenhausgesellschaft, DKG), the German Medical Association (Bundesärztekammer, BÄK), and the Society for Telematic Applications of the Healthcare Card (gematik) which designs and provides the infrastructure for health data services usable with the health card. Currently, the R&D project is in its second phase, covering the three-year period from 2012 to 2014.

Major requirements of the special kind of an electronic health record (EHR) included advanced security and privacy mechanisms and a generic platform semantics that allow for operating various patient-centric e-health applications within the PHR. Another prerequisite was to strongly focus on innovative care scenarios, stable business models, and better patient participation.

2.1 Focus areas

The German health card act (§ 291a SGB V) mentions the electronic health card as an enabler for patient records. The PHR project assesses implementation options and puts the focus on use cases and scenarios that do not solely rely on the defaults of the health card act. That is why additional use cases stretching beyond the health card act with a particular focus on enabling medical research are elaborated, too. The patient record is regarded as a platform for patient centric applications (medication plan, diary etc.). Security aspects play an important role in order to ensure privacy and patient rights. The last thing put into focus is the information model that must support semantic data selection for different applications.

2.2 Key concepts

The PHR project pursuits several key concepts that are introduced subsequently.

2.2.1 Power of disposal of the citizens' data

The basic idea of the PHR is that it is understood as a "tool of the citizen" (the project does not speak of a patient since it is expected that the citizen can store its own data that is outside of a running treatment). The citizen does not regularize the flow of data between systems of health service providers, but initializes and controls it. Essentially, the citizen acts both as a source and target of flows of data and is furthermore a beneficiary of his medical data. The approach taken by the project members does not define a virtual integration of the data, but rather a transmission of copies from medical data into the PHR.

2.2.2 Platform for target group specific record systems

Every citizen has specific needs with regard to a concrete definition of his record. Needs result from his current living situation and mirror in expectations of record features. Vendors should be able to provide target group specific records systems and product variants. This in mind, the PHR has to have support of different interaction and communications patterns (i.e., synchronously and asynchronously communication – an online PHR supports solely synchronously patterns whereas an offline one must support asynchronous communication comparable to email) as well as different authorization mechanisms (ad hoc, in advance, interactively). This offers maximum design flexibility for vendors.

2.2.3 Binding of online records and decentralized data storage

It was recognized that a citizen might want to store data on own decentralized storage devices (e. g., USB flash drives) or rely on secure online storage. In order to meet the individual requirements of the citizens, the project provides a solution for both the binding of online records and decentralized storage devices. The difference is abstracted away from the primary system of the health care provider that accesses the PHR.

2.2.4 Communication of normalized information objects

Besides normalized interfaces, medical data objects need to be interoperable, too. So the last key concept defines a model for a unified handling of different content structures based on Semantic Signifiers developed by the Retrieve, Locate and Update Service (RLUS) standard [RLUS]. This functional standard developed by the Object Management Group (OMG) was chosen to be used on top of a service-oriented security architecture. The introduced level-based concept for step-by-step implementation of normalized content structures (e.g., for a medication summary or a vaccination certificate) relies on the Clinical Document Architecture (CDA) by Health Level Seven (HL7) [HL7 CDA] for the interoperable processing through primary and record systems. Fallback documents (e.g., PDF/A as a standardized version of a Portable Document Format (PDF)) are supported, too.

2.3 Architectural overview

As a design rule, the project idea separates application services from security aspects. This enables reuse of functionalities in other application contexts and provides flexibility for operator models. Security services comprise a central authentication service, a local authorization service, and PKI services. Application services are the mailbox service for communications initiated by the citizen and the communication service that might cache requests from health service providers for offline PHRs.

On the patient side RLUS services encapsulate existing PHR systems (e.g., Microsoft's HealthVault) or simple medical data storage devices (e.g., USB cards). The PHR Core System and the Citizen Client are non-normative and their design/implementation is solely in the responsibility of the respective operators.

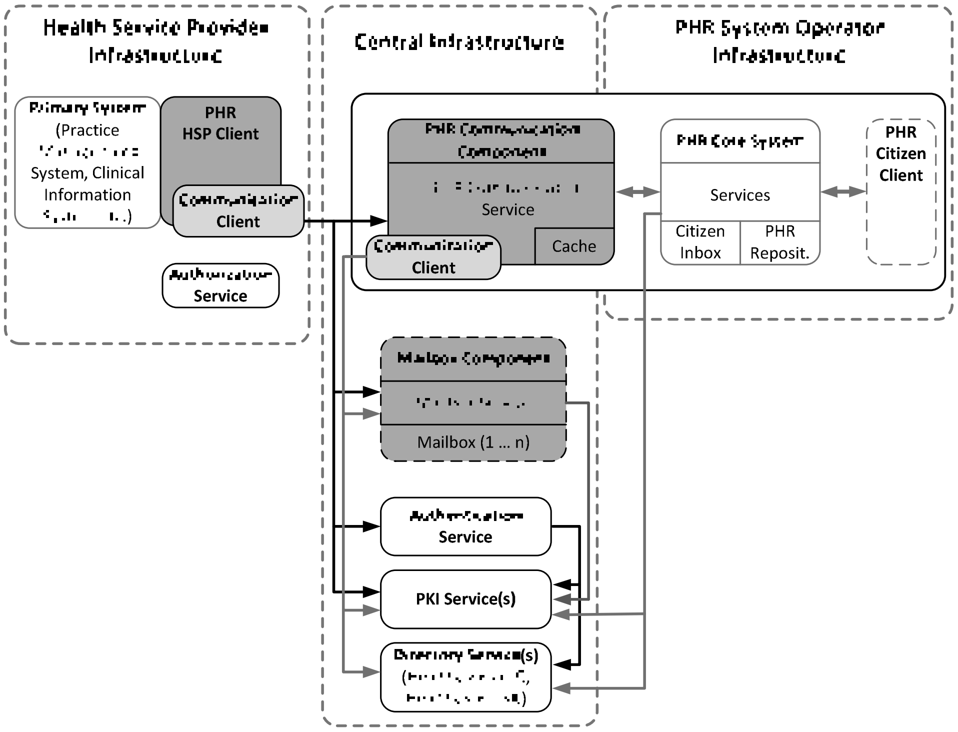

Figure 1: Architectural Overview of German national PHR system

The exchange of medical data is encapsulated in two container formats: Request objects are used to formulate specific information needs. This need is logically expressed as a dedicated object to address the significant decoupling of systems of health service providers and citizens. Request objects must be interpreted by the primary systems. Contrary, a provisioning object represents a logical container for any information provided in the communication flow between the health service provider systems and the PHR. A provisioning object can be regarded as a response to a previously asked information request (synchronous and asynchronous) submitted either by the health service provider or the citizen.

From the health service provider perspective the RLUS operations are safeguarded by SAML assertions [SAML] that carry authentication and authorization data. The architecture allows for both pushing and pulling XACML policies [XACML]. Pushing of policies is used for ad hoc authorization with an electronic health card. In this case an XACML policy is generated on demand by the citizen and pushed in a SAML assertion to the RLUS services within the SOAP security header.

2.3.1 Semantic signifiers

For interoperability reasons, classification and description of the (medical) data must be available in order to ensure the processing in primary systems. A PHR must identify requested data and must be able to transfer local data into interoperable return types. This is due to the fact that the exchange format can differ from the format stored in the PHR – the exchange format may not be exactly the same as the one stored in the PHR. Search requests and filters should point to the specific information model depending on the content type. These demands feature the so-called semantic signifiers specified by RLUS that have an analogy to a WSDL document for a web service description. They describe an underlying information model instead of services. A semantic signifier's essential elements include its name, its description, and a normative data structure that describes instances of it – for example, implementation guidelines, schemas, and specifications for validation.

From the implementation independence view there are no explicit requirements for data storage and information models. However, each vendor must implement functionality for the mapping between internal and external structures defined by semantic signifiers. A further advantage is that any contents and structures might be described. Conversely, each semantic signifier must be explained in detail in order to prevent unchecked distribution.

2.3.2 Capability list

Similar to the capability object that is stored in the inbox of an OBEX (Object Exchange Protocol) server in order to provide capability services for Bluetooth clients, the capability list contains information about the features the PHR provides. This XML-based representation of a capability list includes address information with an Endpoint Reference according to WS-Addressing [WS-Add], a supported public key to be used for hybrid encryption of provisioning objects according to XML Key Management Specification (XKMS) [XKMS], supported communication patterns, and supported semantic signifiers. The capability list is digitally signed by the issuing service.

3 Open eCard App

The Open eCard App is a fully modular and platform-independent implementation of the eCard-API framework [BSI-TR-03112] which can be further extended by plug-ins and add-ons. The architecture of the Open eCard App is depicted in Figure 2.

Access to card terminals, smart cards and the functions they make available is provided by the following three components of the Open eCard-Framework:

- The *Interface Device (IFD)* module provides a generalized interface to access arbitrary card terminals and smart cards and an interface for password-based authentication protocols (e.g., PACE [BSI-TR-03110]). It is specified in part 6 of [BSI-TR-03112] and part 4 of [ISO24727].

- The *Event Manager* monitors the IFD for incoming events (e.g., insertion or removal of smart cards) and triggers the card recognition. Registered components are notified by the Event Manager if an event occurs.
- The *Service Access Layer (SAL)* manages all available smart cards and executes all kinds of protocols (e.g., the Extended Control Access protocol [BSI-TR-03110]). It is specified in part 4 of [BSI-TR-03112] and part 3 of [ISO24727].

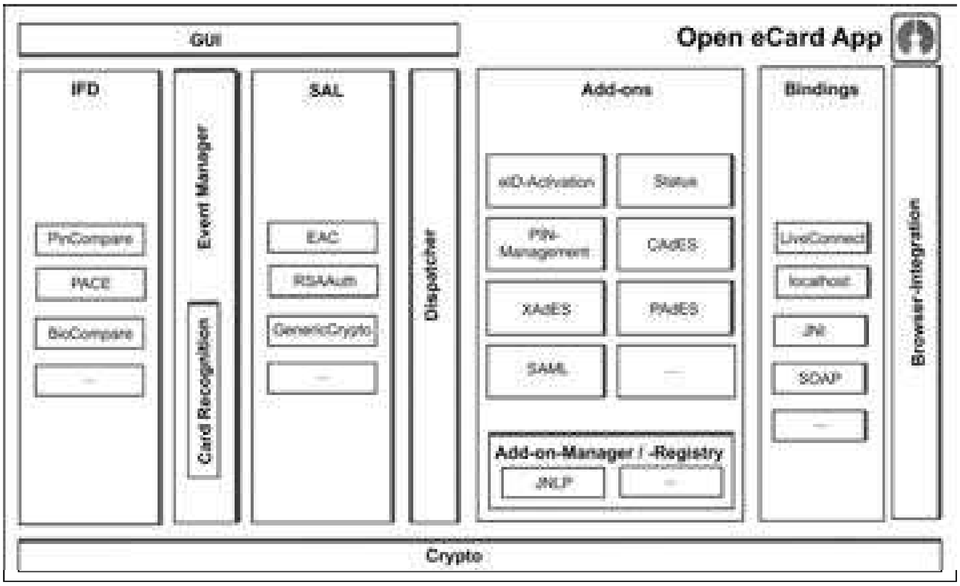

Figure 2: Extensible architecture of the Open eCard App

The Open eCard platform provides different extension points to support newly issued smart cards, new protocols and new application specific functions. A detailed description of the architecture as well as the plug-in mechanism of the Open eCard platform can be found in [WHP+13].

The PHR plug-in described in the next section will basically utilise the SAL for all operations that include access to the electronic health card, support the establishment of TLS channels with client authentication using the health card and provide a simple application interface which allows to access the plug-in via the localhost binding. In this binding, which has been introduced in part 7 of [BSI-TR-03112] for purposes of eID-activation, a function is called by performing an HTTP GET request to http://127.0.0.1/ (localhost) and parameters may be provided in URL-encoded form. See [WHP+13] for more information on application plug-ins and corresponding binding mechanisms.

4 Approach

In order to demonstrate the feasibility of using the Open eCard platform for health care applications, we consider the following scenario as depicted in Figure 3:

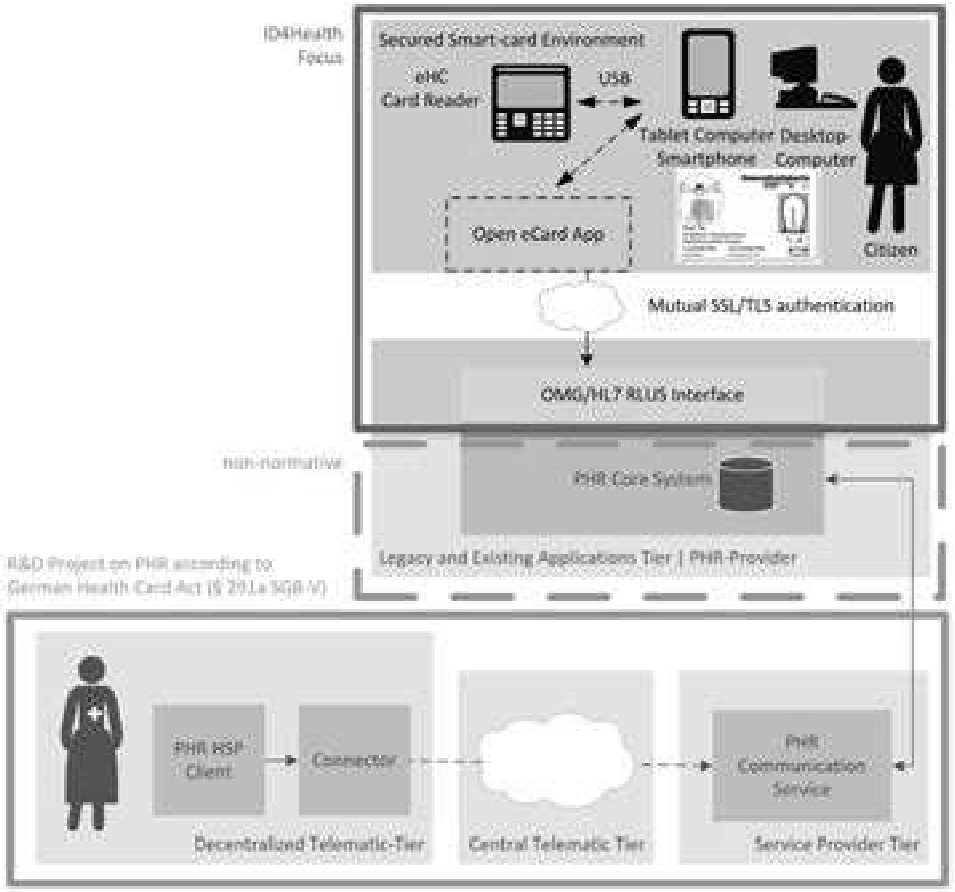
ID4Health Focus
Secured Smart-card Environment
eHC Card Reader
USB
Tablet Computer Smartphone
Desktop-Computer
Citizen
Open eCard App
Mutual SSL/TLS authentication
OMG/HL7 RLUS Interface
non-normative
PHR Core System
Legacy and Existing Applications Tier | PHR-Provider
R&D Project on PHR according to German Health Card Act (§ 291a SGB V)
PHR HSP Client
Connector
PHR Communication Service
Decentralized Telematic-Tier
Central Telematic Tier
Service Provider Tier

Figure 3: Solution scenario for using the Open eCard App to access the PHR system

A citizen wants to access his PHR via his desktop computer or mobile device. The citizen has his electronic health card and a smart card reader, which is connected by means of USB for example.

The PHR Core System is operated in a secure environment by a PHR provider. The environment can be accessed via an RLUS interface over a secure point-to-point data channel. This data channel is established by a mutual authentication which is performed using the PHR provider's certificate and the X.509-certificate (C.CH.AUTN) deployed with the citizen's health card.

4.1 Architecture

The linkage of a citizens PHR application with the PHR Core System is realized by using the Open eCard platform and developing a plug-in which

1. uses the smart-card functionality and cryptographic features of the framework and
2. provides a PHR-specific application interface, which can be used by application services.

The following layer model depicts the relationship of the logical components that ensure the secure communication of PHR applications and PHR core system using the extended Open eCard App.

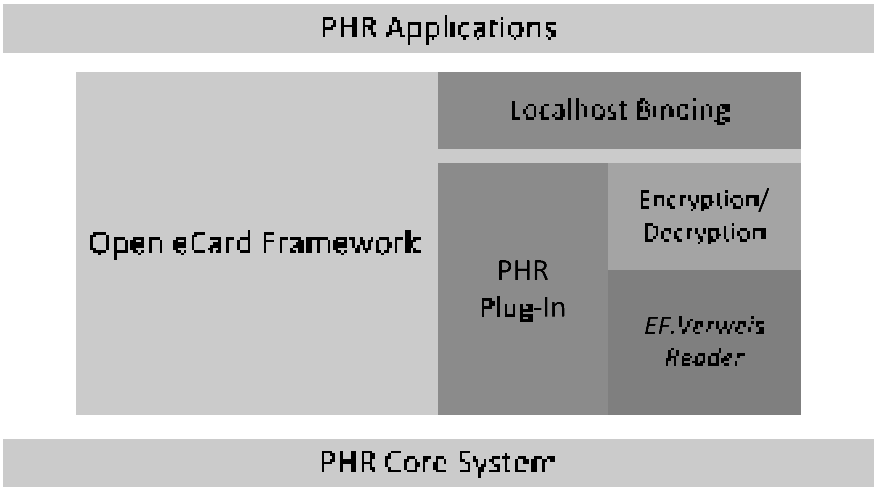

Figure 4: Accessing the PHR Core System with the extended Open eCard App

The system comprises three main components:

1. PHR Application, which provide tailormade user interfaces
2. Open eCard Framework with PHR specific plug-in
3. PHR Core System, which in particular contains the repository service.

The PHR Core System provides the PHR client with personal medical data. The PHR client is realized by a PHR-specific plug-in for the Open eCard platform. It has an interface following the RLUS standard [RLUS] and exchanges messages via a mutually authenticated TLS channel.

Third-party applications can utilise the PHR plug-in to store and retrieve information from the PHR Core System. For this purpose the PHR plug-in provides two operations: the `getInformationObject` function for information retrieval and the `putInformationObject` function for information storage. Both functions are available through the localhost binding of the Open eCard framework.

The information retrieval from the PHR Core System works as follows:

A third-party application calls the `getInformationObject` function with a parameter identifiying a certain semantic signifier that classifies the requested information. If the electronic health card is inserted in the card terminal the personal identification number is requested in order to read a special container (EF.Verweis) on the card that stores the service reference to the PHR Core System. Otherwise the citizen will be prompted to insert the smart-card. The service reference includes a provider ID as well as a record ID. The provider ID can be used to determine the service location of the PHR Core System. The plug-in establishes a mutually authenticated TLS channel to the PHR Core System using the C.CH.AUTN certificate, which is an X.509 certificate stored on the health card carrying a pseudonym of the citizen. The capability list of the PHR is initially requested which includes the supported record types (i.e., semantic signifiers) as well as a the public key of the record instance which can be used to protect symmetric document keys that in turn encrypt information objects submitted by the PHR plug-in to the PHR Core System. Afterwards, the specified content, addressed by a parameter of the `getInformationObject` function, is requested from the PHR Core System using the RLUS LIST operation which returns the requested data as an encrypted provisioning object. The plug-in decrypts the retrieved provisioning object by means of the health card's key material and returns it to the calling application as a Base64-encoded string.

To store information in the PHR Core System, the following steps are performed:

A third-party application invokes the `putInformationObject` function with the Base64-encoded information object and a semantic signifier. The plug-in decodes the information object and generates a symmetric document key. In order to secure the information object, the plug-in encrypts it by means of that document key. The document key is secured using the public key of the capability list (which was retrieved by a former call). Finally, both the (symmetrically) encrypted information object and the associated encrypted (symmetric) data key are used to create a provisioning object which is embedded as a payload in a RLUS PUT message and sent to the PHR Core System in order to store the information object.

The Open eCard App implements all cryptographic components and integrates smart card terminals, which may or may not include a display and key pad for secure PIN entry. The PHR plug-in mirrors a much simpler version of the PHR Core System interface to the PHR client-application and provides high-level operations for encryption and signing based on the smart card currently in use.

5 Summary and conclusion

The present paper briefly described how the extensible Open eCard platform can be used for securely accessing sensitive medical information, which is stored in the German national Personal Health Record according to § 291a SGB V. This demonstrates not only the feasibility of the ID4health approach but also underlines the power of the extensibility of the Open eCard platform. As the supported extension mechanisms are both independent of the used smart card and the implemented application logic, it may be expected that this approach can easily be carried over to entirely different application scenarios.

Future work will include the support of record initialization (including key generation and secure transmission) as well as the investigation of the complementary use of the identity card for health professionals in consideration of signing and decrypting medical data. This is very useful when the Open eCard plug-in is deployed in primary systems of health service providers. Such scenarios require e.g. the issuing of authorization assertions that are digitally signed by the electronic health card (named ad-hoc authorization). Hence, the access control system of the PHR Core system can verify the eligible use of the patient data by health professionals.

References

[BSI-TR-03110] Bundesamt für Sicherheit in der Informationstechnik (BSI): Advanced Security Mechanism for Machine Readable Travel Documents – Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI), Technical Directive TR-03110, Version 2.10, 2012

[BSI-TR-03112] Bundesamt für Sicherheit in der Informationstechnik (BSI): eCard-API-Framework. Technical Directive TR-03112, Version 1.1.2, Part 1-7, 2012

[HL7 CDA] ISO/HL7 27932:2009: Data Exchange Standards - HL7 Clinical Document Architecture, Release 2

[HPS+12]	Hühnlein, D.; Petrautzki, D.; Schmölz, J.; Wich, T.; Horsch, M.; Wieland, T.; Eichholz, J.; Wiesmaier, A.; Braun, J.; Feldmann, F.; Potzernheim, S.; Schwenk, J.; Kahlo, C.; Kühne, A.; Veit H.: On the design and implementation of the Open eCard App, In Sicherheit 2012, GI-LNI (2012). http://subs.emis.de/LNI/Proceedings/Proceedings195/95.pdf
[HoSt11]	Horsch, M.; Stopczynski, M.: The German eCard-Strategy, CDC Report, TU Darmstadt, February 2011, https://www.cdc.informatik.tu-darmstadt.de/en/publikations-details/?no_cache=1&tx_bibtex_pi1%5Bpub_id%5D=TUD-CS-2011-0179
[ISO24727]	ISO/IEC 24727: Identification cards – Integrated circuit cards programming interfaces, Part 1-6, International Standard, 2008
[RLUS]	OMG: Retrieve, Locate, and Update Service (RLUS) Specification, Version 1.0.1., 2011.
[SAML]	Cantor, S.; Kemp, J.; Philpott, R.; Maler, E.: Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0., OASIS Standard, 15.03.2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, 2005
[TLS]	Dierks, T.; Rescorla, E.: The Transport Layer Security (TLS), Protocol Version 1.2. Request For Comments – RFC 5246. http://www.ietf.org/rfc/rfc5246.txt, August 2008
[WHP+13]	Wich, T.; Horsch, M.; Petrautzki, D.; Schmölz, J.; Hühnlein, D.; Wieland, T.; Potzernheim: An extensible platform for eID, signatures and more, in the present proceedings
[WS-Add]	Gudgin, M.; Hadley, M; Rogers, T.: Web Services Addressing 1.0 – Core, 2006, http://www.w3.org/TR/ws-addr-core/
[XACML]	Moses, T.: XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) Version 2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf
[XKMS]	Hallam-Baker, P.; Mysore, S. H.: XML Key Management Specification (XKMS 2.0), http://www.w3.org/TR/xkms2/