How to contribute as product vendor?

The non-profit Open Signature Initiative aims at supporting the process of implementing a single European market of trustworthy services by enhancing transparency and interoperability with respect to electronic signature technology and related trust services.

Vendors of signature creation devices, signature application components and services are cordially invited to provide ***technical information*** about their products as well as corresponding ***test and demo versions*** of the components or services, which can be evaluated by interested parties to verify the interoperability claims.

1. How should the product related information be provided?

The description of the products should be provided in form of an XML-based `Product`-element according to the schema available at http://ws.openecard.org/schema/OpenSignature.xsd and the description provided in Section 3 below.

The XML-based description and any related inquiries should be sent to signature@openecard.org.

2. The Open Signature Component Model

Based on international standards and existing products the "Open Signature Component Model" has been developed.

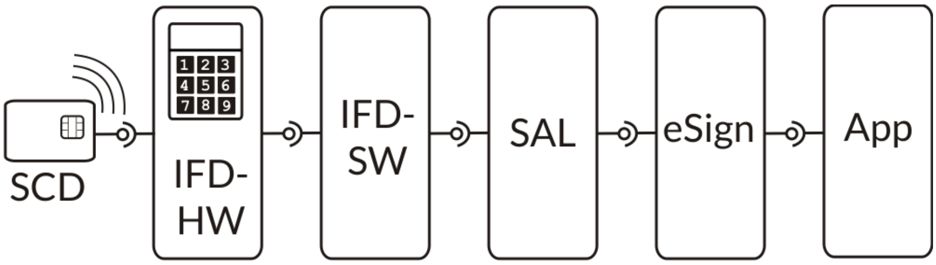

Figure 1: The "Open Signature Component Model"

As depicted in Figure 1 this abstract model distinguishes the following modules:

- *SCD* – is the Signature Creation Device (SCD), which may be realized as smart card or other secure element which is equipped with a contact-based or contactless interface and may support the commands specified in [1] and [2] for example.
- *IFD-HW* – is the hardware-based part of the Interface Device (IFD) which allows to transmit commands from the host components (IFD-SW, SAL, eSign, App) to the SCD.
- *IFD-SW* – is the software-based part of the interface device, which may be equipped with an interface according to [3] for example.
- *SAL* – is the Service Access Layer (SAL), which in particular may generate smart card commands, which are transported to the SCD via the IFD-SW and IFD-HW module. This module may be equipped with an interface according to [4] for example.
- *eSign* – is the module, which may allow to generate and/or verify advanced electronic signatures according to [5], [6] and [7] for example.
- *App* – is the application module, which may provide additional application logic.

It should be noted that the "Open Signature Component Model" is an abstract model, which may be implemented in various ways. In particular the model aims at covering different system architectures, including classical rich client systems with local smart card access as well as different flavours of client-server architectures, which may involve mobile devices and cloud services for example.

Please send a mail to signature@openecard.org if you feel that an important feature or standard is missing in the XML-schema.

3. The structure of the `Product`-element

The `Product`-element has the following child elements:

- `OrgInfo` – provides informatioin about the originator, vendor or provider of the component or service. Details with respect to this element are provided in Section 3.1 below.
- `ProductInfo` – provides information about the component or service. Details with respect to this element are provided in Section 3.2 below.
- `AdditionalInformation` - may appear multiple times in order to provide additional information about the product.

3.1 The structure of the `OrgInfo`-element

The `OrgInfo`-element has the following child elements:

- `OrgLink` – is a localised link to the web site of the organisation which offers the component or service. If there are different web sites in different languages, this element may appear multiple times with different `xml:lang` attributes.
- `OrgName` – is the localised name of the organisation which offers the component or service. If there are different names for the organisation in different languages, this element may appear multiple times with different `xml:lang` attributes.
- `OrgLogo` – is a link to the logo of the organisation who offers the component or service.
- `OrgContact` – may appear multiple times and contain (pointers to) contact information.

3.2 The structure of the `ProductInfo`-element

The `ProductInfo`-element has the following child elements:

- `ProductName` – is the localised name of the offered component or service. If there are different names for the product in different languages, this element may appear multiple times with different `xml:lang` attributes.
- `ProductClass` – indicates to which classes the product belongs. It may appear multiple times, whereas the following values are specified:
    - `Library` – indicates that the product is available as library, which can be integrated into other applications.
    - `MobileApp` – indicates that the product is available as application for a mobile device.
    - `RichClient` – indicates that the product is available as standalone client application for some PC-based platform.
    - `Applet` – indicates that the product is available as Java-applet, which can be executed within a browser.
    - `ThinClient` – indicates that the product is or contains a thin client application, which needs to communicate with a corresponding server system to create or verify electronic signatures.
    - `Server` – indicates that the product is or contains a server system.
    - `Service` – indicates that the offering is a service for the creation or verification of electronic signatures.

- `ProductLogo` – may point to the logo of the product.
- `ProductLink` – is a localised link to the product site, which should allow to obtain test and demo versions of the product. If there are multiple sites for different languages, this element may appear multiple times with different `xml:lang` attributes.
- `Version` – indicates the version of the product.
- `License` – indicates the license of the product or the terms and conditions of an offered service. If the product is available under more than one license, this element may appear multiple times. The following values are specified:
  - `proprietary` – specifies that the product is distributed under a proprietary license. In this case there should be a pointer to additional information in the `LicenseLink` element below.
  - `AGPL-3.0` - http://opensource.org/licenses/AGPL-3.0
  - `Apache-2` – http://opensource.org/licenses/Apache-2.0
  - `BSD-2` – http://opensource.org/licenses/BSD-2-Clause
  - `BSD-3` – http://opensource.org/licenses/BSD-3-Clause
  - `CDDL-1.0` – http://opensource.org/licenses/CDDL-1.0
  - `EPL-1.0` – http://opensource.org/licenses/EPL-1.0
  - `EUPL-1.1` - http://opensource.org/licenses/EUPL-1.1
  - `GPLv2` – http://opensource.org/licenses/GPL-2.0
  - `GPLv3` – http://opensource.org/licenses/GPL-3.0
  - `LGPLv2.1` – http://opensource.org/licenses/LGPL-2.1
  - `LGPLv3.0` – http://opensource.org/licenses/LGPL-3.0
  - `MIT` – http://opensource.org/licenses/MIT
  - `MPL-2.0` – http://opensource.org/licenses/MPL-2.0
  - `otherOS` – specifies that the product is distributed under some other open source license. In this case there should be a pointer to additional information in the `LicenseLink` element below.
- `LicenseLink` – is an optional element, which points to more information with respect to the license of the product or the terms and conditions of an offered service. This element should be available, if the license element above equals `proprietary` or `otherOS`.
- `AppFeatures` – may appear multiple times and should decsribe the features of the App-module of the product (cf. Figure 1).
- `eSignFeatures` – describes the features of the eSign-module (see Figure 1). This element contains the following child elements:
  - `API` – describes the set of standardised APIs, which are supported by the eSign-module, whereas the following values are specified:
    - `OASIS-DSS-Core` – indicates that the a subset of the OASIS DSS Core API specified in [8] is supported.

        - `OASIS-DSS-AdES` – indicates that the API specified in the AdES-specific profile according to [9] is supported.
        - `OASIS-DSS-VR` – indicates that the product supports the creation of signature verification reports according to [10].
        - `BSI-TR-03112-2` – indicates that the API specified in [11] is supported.
        - `BSI-TR-03125-S.4` – indicates that the S.4 interface according to [12] is supported.
    - `Format` – describes the standardized formats supported by the eSign-module, whereas the following values are specified:
        - `CAdES` – indicates that the CAdES-profile according to [5] and [13] is supported.
        - `XAdES` – indicates that the XAdES-profile according to [6] and [13] is supported.
        - `PAdES` - indicates that the PAdES-profile according to [7] and [13] is supported.
        - `ASiC` – indicates that the ASiC container format according to [14] is supported.
        - `RFC4998` – indicates that ASN.1-based evidence record format according to [15] are supported.
        - `RFC6283` – indicates that XML-based evidence record format according to [16] are supported.
    - `Other` – may appear multiple times in order to describe other features of the eSign-module.
- `SALFeatures` - describes the features of the SAL-module (see Figure 1). This element contains the following child elements:
    - `API` – describes what kind of standardized APIs are supported by the SAL-module, whereas the following values are specified:
        - `ISO24727-3` – indicates that the a subset of the SAL-API specified in [4] is supported.
        - `PKCS11` – indicates that a subset of the API specified in [17] is supported.
        - `MS-CAPI` – indicates that a subset of the API specified in [18] is supported.
    - `Device` – may be present multiple times in order to describe the set of supported eID tokens or signature creation devices. For cards with existing CardInfo-file according to [4] the corresponding `ShortName` should be inserted here.
    - `DIDProtocol` – may be present multiple times in order to specify the standardised authentication protocols according to [4], which are supported by the SAL.

    - `Other` – may appear multiple times in order to describe other features of the SAL-module.
- `IFDSWFeatures` - describes the features of the IFD-SW-module (see Figure 1). This element contains the following child elements
    - `API` – may be present multiple times in order to describe the supported programming interfaces of the IFD-SW-module, whereas the following values are specified:
        - `ISO24727-4` - indicates that the IFD-SW-module supports a subset of the IFD-API specified in [3].
        - `PCSC` – indicates that the IFD-SW-module supports card terminals, which provide a PC/SC driver according to [19].
        - `OpenMobile` – indicates that the IFD-SW-module supports secure elements, which are accessible via the Transport API of the Open Mobile API [20].
        - `CT-API` – indicates that the IFD-SW-module supports card terminals via the classical CT-API-interface [21].
        - `SICCT` – indicates that the IFD-SW-module supports card terminals via the SICCT-interface [22].
- `IFDHWFeatures` - describes the features of the IFD-HW-module (see Figure 1). This element contains the following child elements
    - `HostInterface` – may be present multiple times in order to describe the supported interfaces of the IFD-HW-module offered to the host, whereas the following values are specified:
        - `USB` - indicates that the IFD-HW-module is accessible via USB.
        - `RS-232` - indicates that the IFD-HW-module is accessible via the serial RS-232 interface.
        - `Bluetooth` – indicates that the IFD-HW-module is accessible via Bluetooth.
        - `PCSC` – indicates that the IFD-HW-module provides a PC/SC driver according to [19].
        - `CT-API` – indicates that the IFD-HW-module is accessible via the classical CT-API-interface [21].
        - `SICCT` – indicates that the IFD-HW-module is accessible via the SICCT-interface [22].
    - `DeviceInterface` – may be present multiple times in order to describe the supported interfaces of the IFD-HW-module for connecting to the signature generation devices, whereas the following values are specified:
        - `ISO7816-3` – indicates that the IFD-HW module supports SCDs with a contact based interface [23].

- `ISO14443` – indicates that the IFD-HW module supports SCDs with a contactless interface according to ISO/IEC 14443 [24].

  - `IFDFeatures` – may be present multiple times in order to describe the features of the IFD-HW module, whereas the following values are specified:
    - `PinPad` – indicates that the IFD-HW module has a PIN pad for secure PIN entry.
    - `Display` – indicates that the IFD-HW module has a display.
    - `SAM` – indicates that the IFD-HW module may be equipped with a Secure Access Module (SAM).
    - `Fingerprint` – indicates that the IFD-HW module is equipped with a fingerprint sensor.
  - `Other` – may appear multiple times in order to describe other features of the IFD-HW module.
  - `SCDFeature` – may appear multiple times in order to describe the features of a signature creation device, whereas the following values are defined:
    - `EN14890` – indicates that the SCD supports the interface defined in [1] and [2].
    - `ISO7816-3` – indicates that the SCD has a contact based interface [23].
    - `ISO7816-4` – indicates that the SCD supports standardized APDUs according to [25].
    - `ISO7816-15` – indicates that the features of the SCD are described by an Cryptographic Information Application according to [26].
    - `ISO14443` – indicates that the SCD has a contactless interface according to ISO/IEC 14443 [24].
    - `ISO24727-2` – indicates that the SCD supports the generic card interface according to [27].
- `OtherFeature` – may appear multiple times in order to describe other features of the product.
- `Certification` – may appear multiple times in order to describe certification-related aspects of the product. This element contains the following child elements:
  - `Type` – indicates the type of certification which has been obtained.
  - `Date` – indicates the date of the certification.
  - `Body` – specifies the certifiation body, who provided the certification.

- `InfoLink` – may appear multiple times and provide links to additional information related to the certification (e.g. security target, certification report).
- `SupportedPlatform` – may appear multiple times in order to specify the supported platforms of the product.

  This element should describe the supported platforms as precisely as possible. This may involve providing information about the supported operation system (e.g. Windows {XP, Vista, 7, 8, Mobile, Server *xy* etc. }, Linux, MacOSX, iOS, Android, Blackberry, etc.), the variant (e.g. support for Terminal Server), the distribution (e.g. Debian, etc.), the runtime environment (e.g. if it requires a specific version of the Java runtime environment) and the version.

References

[1] European Norm 14890-1, *Application Interface for smart cards used as Secure Signature Creation Devices -- Part 1: Basic Services,* 2009.

[2] European Norm 14890-2, *Application Interface for smart cards used as Secure Signature Creation Devices -- Part 2: Additional Services,* 2008.

[3] ISO/IEC 24727, *Identification cards - Integrated circuit cards programming interfaces - Part 4: API Administration,,* 2008.

[4] ISO/IEC 24727, *Identification cards - Integrated circuit card programming interfaces - Part 3: Service Access Layer,* International Standard, 2008.

[5] ETSI, *CMS Advanced Electronic Signatures (CAdES),* ETSI TS 101 733, Version 1.8.1, 2009.

[6] ETSI, *Technical Specification XML Advanced Electronic Signatures (XAdES),* ETSI TS 101 903, Version 1.4.1, 2009.

[7] ETSI, *PDF Advanced Electronic Signature Profiles (PAdES),* ETSI TS 102 778, Part 1 - 5, 2009.

[8] S. Drees, *Digital Signature Service Core Protocols, Elements, and Bindings,* Version 1.0, OASIS Standard, http://docs.oasis-

open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf.

[9] J. C. Cruellas, *Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0,* Version 1.0, OASIS Standard, http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf, 2007.

[10] D. Hühnlein, *Profile for comprehensive multi-signature verification reports for OASIS Digital Signature Services Version 1.0,* OASIS Committee Specification 01, http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.pdf, 2009.

[11] Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik), *eCard-API-Framework - eCard-Interface,* https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03112/api1_teil2_pdf.pdf?__blob=publicationFile, 2012.

[12] Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnik), *Preservation of Evidence for cryptographically signed documents,* https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG03125/TG-03125AnnexTR-ESOR-E.pdf?__blob=publicationFile, 2011.

[13] *Commission decision of 25 February 2011 establishing minimum requirements for the cross-border processing of documents signed electronically by competent authorities under Directive 2006/123/EC (...) on services in the internal market,* 2011/130/EC, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:053:0066:0072:EN:PDF, 2011.

[14] ETSI, *Electronic Signatures and Infrastructures (ESI); ASiC Baseline Profile,* ETSI TS 103 174 V1.1.1 (2011-09), 2011.

[15] T. Gondrom, R. Brandner and U. Pordesch, *Evidence Record Syntax (ERS),* IETF RFC 4998, 2007.

[16] A. J. Blazic, S. Saljic and T. Gondrom, *Extensible Markup Language Evidence Record Syntax (XMLERS),* IETF RFC 6283, 2011.

[17] RSA Laboratories, *PKCS #11: Cryptographic Token Interface Standard - Version 2.30,* http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-11-cryptographic-token-interface-standard.htm, 2009.

[18] Microsoft Inc., *Cryptography Reference*, http://msdn.microsoft.com/en-us/library/aa380256.aspx.

[19] PC/SC Workgroup, *PC/SC Workgroup Specifications 2.0*, 2005.

[20] simalliance, *Open Mobile API specification, Version 2.04,* http://www.simalliance.org/en?t=/documentManager/sfdoc.file.supply&fileID=1375105650166.

[21] J. Attrott, L. Eckstein, B. Kowalski, R. Moos, H. Reimer and B. Struif, *Anwendungsunabhängiges CardTerminal-Application Programming Interface für Chipkartenanwendungen (CT-API),* 2001.

[22] TeleTrusT, *SICCT-Spezifikation*, 2006.

[23] ISO/IEC 7816-3, *Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols,* International Standard, 2006.

[24] ISO/IEC 14443, *ISO/IEC 14443: Identification cards - Contactless integrated circuit(s) cards - Proximity cards - Part 1-4*, International Standards, 2001.

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

[26] ISO/IEC 7816-15, *Identification cards - Integrated circuit cards - Part 15: Cryptographic information application,* International Standard, 2004.

[27] ISO/IEC 24727-2, *Identification cards - Integrated circuit cards programming interfaces - Part 2: Generic Card Interface,* International Standard, 2008.