Architecture of ID-software

Version 0.4

ID-software version: 3.9

Document versions

Version	Date	Author	Comments
0.1	12.05.2014	Kristi Uukkivi	Created initial structure of the document
0.2	15.09.2014	Kristi Uukkivi	Added content
0.3	22.09.2014	Kristi Uukkivi	Added content
0.4	14.11.2014	Kristi Uukkivi	Changes according to feedback. Changes in layout, diagrams, etc

Table of contents

1 Introduction

The purpose of this document is to describe the architecture of ID-software.

ID-software is a collection of software components offering support for PKI-based functionality, i.e. operations with different cryptographic tokens (e.g. eID cards), handling digitally signed documents, file encryption/decryption and signing and authentication in web environment. The ID-software comprises end-user applications, software libraries, web components, drivers for communicating with the cryptographic tokens and other complementary components.

This document covers description of ID-software and its components, their deployment in different environments, provided and required interfaces. The document does not include components that have reached the end of their support nor the components that have not yet been released[1].

The document is based on the latest released state of the ID-software components. At the time of writing, the latest released version of ID-software is **version 3.9**. Latest version numbers of the various ID-software components are provided in [6].

The document is targeted for:

1. Owners/managers of the software;
2. Contractors;
3. Contributors interested in developing ad-hoc solutions;
4. Integrators/software developers interested in integrating the software with third-party information systems;
5. International audience – contributors/integrators from countries other than Estonia who wish to use the software internationally and/or contribute in its development.

[1] Except of DigiDoc3 Client's international release (v3.9.7) which will be included after its incorporation with DigiDoc3 Client's Estonian version.

2 Background

The main owner/manager of the ID-software is Estonian Information System Authority (RIA, https://www.ria.ee/en/).

Main contractor for developing the software is AS Sertifitseerimiskeskus (SK, https://sk.ee/en). In case of a few of the components, SK is also the owner.

Development of ID-software has been mainly done in Estonia, however, the ID-software is released for international usage.

The software is distributed open-source (under LGPL/BSD licence) and is accessible from the following locations:

1. GitHub repository for the source code (https://github.com/open-eid). Some of the components are yet to be added in the future.
2. Release repository for binaries: https://installer.id.ee/

ID-software components can be logically divided in the following groups:

1. **Desktop applications** for end-users;
2. **Software libraries** for integrators/software developers to integrate the libraries' functionality with third-party information systems/applications;
3. **Web components** for integrators/software developers to add the signature creation and authentication functionality in web environment to third-party web applications;
4. **Drivers** for communication with the cryptographic tokens that conduct the PKI operations;
5. **Other (supportive) components** for packaging, installation, updating and version-control of the software.

The following table maps the ID-software components, their owner/developer (i.e. the main contractor) and the functionality they offer.

Table 1. Mapping of ID-software components and functions

	Component	Function					Owner/ Developer
		Handling DDOC/ BDOC documents	Handling CDOC documents	Calculating RSA signature	Card management operations	Authenti-cation	
Desktop applications	**DigiDoc3 Client**	+*	-	-	-	-	RIA/SK
	DigiDoc3 Crypto	-	+	-	-	-	RIA/SK
	ID-card utility	-	-	-	+*	-	RIA/SK
Software libraries	**JDigiDoc (Java)**	+	+	+*	-	-	RIA/SK
	Libdigidocpp (C++)	+	-	+*	-	-	RIA/SK
	CDigiDoc (C)	+	+	+*	-	-	RIA/SK
	NDigiDoc (.NET)	-	+	-	-	-	SK/SK
Web components	**Browser signing modules**	-	-	+*	-	-	RIA/SK
	js-token-signing (JavaScript)	-	-	+*	-	-	SK/SK
	pkcs11-module-loader	-	-	-	-	+**	RIA/SK
Driver components	**Minidriver**	-	-	+*	-	+*	RIA/SK
	EstEID-pkcs11	-	-	+*	-	+*	RIA/SK
	EstEID-tokend	-	-	+*	-	+*	RIA/SK
	Smartcardpp	-	-	+*	+*	+*	RIA/SK

* - The functionality is provided via base components

** - The component is used only once for setting the proper parameters for authentication in Firefox browser.

The main functions offered by ID-software are described in the following table.

Table 2. Functions offered by ID-software

Function	Description
Handling DDOC/BDOC documents	Handling documents in BDOC 2.1 [1] and DIGIDOC-XML 1.3 (DDOC) [3] digital signature formats that are profiles of ETSI XAdES (XML Advanced Electronic Signature, [4]) format. More information on the formats' life cycle can be found from [2].
Calculating RSA signature	Calculating the RSA signature value in browser or desktop/server environment. The operation involves connecting with the signature token's driver, sending the data to be signed and receiving digital signature value calculated with the token owner's RSA private key. The following cryptographic tokens are supported: 1. Hardware-based tokens (e.g. PKCS#11-based eID cards, USB cryptostick and Mobile-ID) 2. Software-based tokens (e.g. PKCS#12 software token)
Handling CDOC documents	Encrypting and decrypting documents in ENCDOC-XML 1.0 (CDOC) format ([5]).
Card management operations	Renewal of the certificates on the card, PIN/PUK management, reading personal data file.
Authentication	Authentication with ID-card. The operation is generally done via native operating system/browser components. In case of Estonian ID-cards and Firefox browser, a PKCS#11 module loader script is used for setting the proper parameters for authentication in Firefox browser.

3 Component model

The following chapter depicts ID-software component diagrams, including variations of the components used in different supported environments.

In the context of the component diagrams in this document, the ID-software components have been divided to three different packages to show the component's owner/developer:

1. Components of ID-software that are owned by RIA and developed by SK: placed in "RIA/SK" package, marked with pink colour;
2. Components of ID-software that are owned and operated by RIA: placed in "RIA" package, coloured with purple;
3. Components of ID-software that are owned and developed by SK: placed in "SK" package, coloured with light-blue.

The component model also includes components that are external to ID-software (marked with grey colour).

Note that not all of the external base libraries are included in the component model to avoid duplicity with other documentation – the base libraries are listed and described in the documentation of the respective ID-software components and can be accessed via the references provided.

3.1 Desktop applications

3.1.1 DigiDoc3 Client

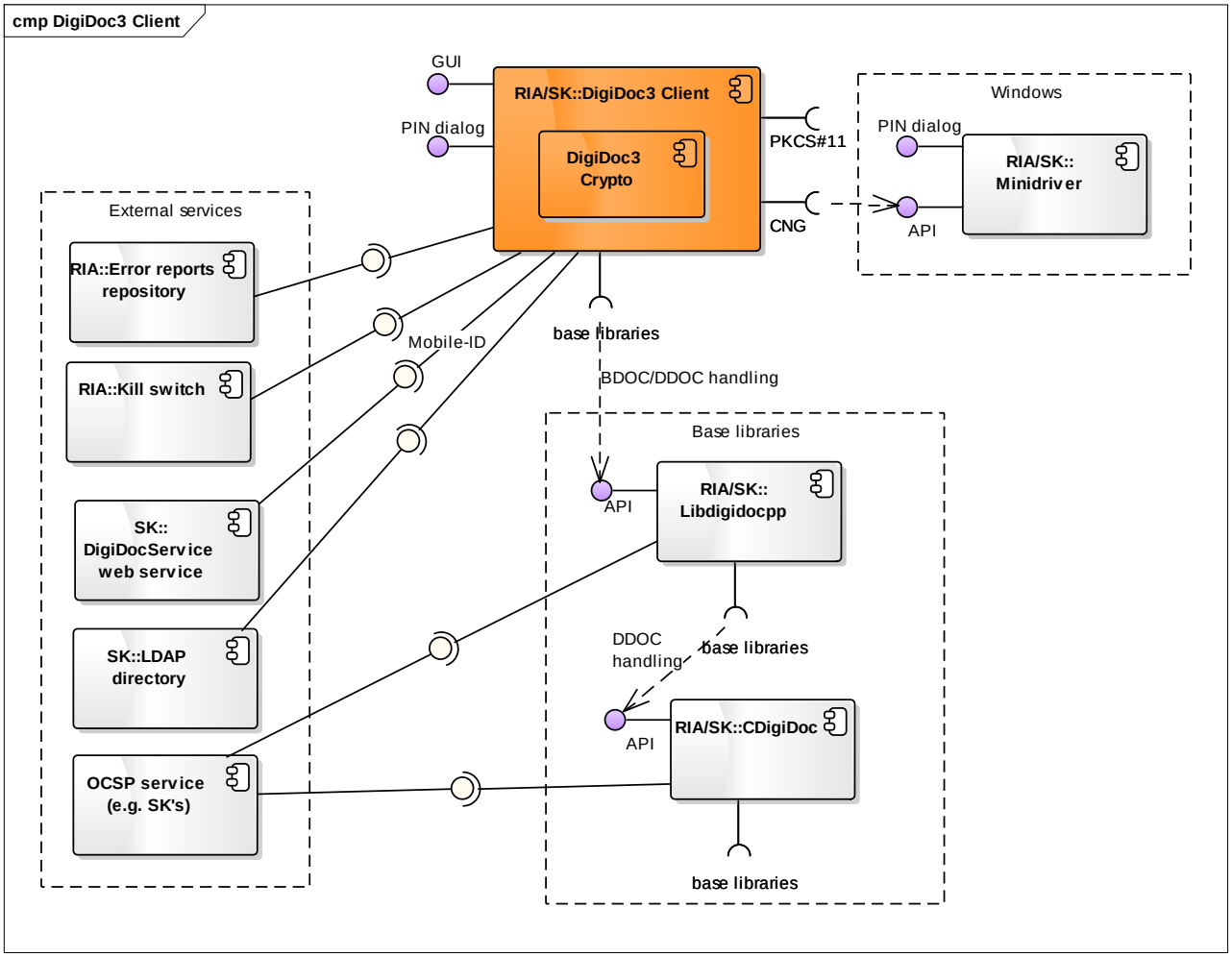

Figure 1. Components of DigiDoc3 Client

Table 3. Components of DigiDoc3 Client

Component	Description	Owner/ Developer
DigiDoc3 Client	End-user desktop applications that own a common GUI. DigiDoc3 Client enables handling digitally signed documents. DigiDoc3 Crypto subcomponent enables file encryption/decryption.	RIA/SK
DigiDoc3 Client base libraries	See DigiDoc3 Client documentation (to be included in the distribution package).	-
Error reports repository	Repository where the DigiDoc3 Client application's and ID-card utility program's error reports (generated with BreakPad base library) are sent.	RIA
Kill switch	Service for centrally managing DigiDoc3 Client application's life cycle. The application periodically connects with the service to check if the application's version is still supported. If not, then the application cannot be used any longer and a newer version must be installed.	RIA
DigiDocService web service	SOAP-based web service that is used by DigiDoc3 Client for signature creation with Mobile-ID. See also [12].	SK/SK

Component	Description	Owner/ Developer
LDAP directory	Directory of active certificates issued by SK (as the CA in Estonia). The directory is used by DigiDoc3 Crypto subcomponent for finding authentication certificate (and the respective public key) of the recipient of the encrypted document. See also [13].	SK/SK
OCSP service	RFC2560-based OCSP service. Also offered by SK for Estonian and a number of foreign certificates (see www.sk.ee).	-
Libdigidocpp	Described in chap. 3.1.2.1	RIA/SK
CDigiDoc	Described in chap. 3.1.2.1	RIA/SK
Minidriver	Used via CNG interface in Windows environment only. Described in chap. 3.4	RIA/SK

3.1.1.1 Interfaces

Provided:

1. Graphical user interface - interface for handling BDOC [1], DDOC [3], CDOC [5] documents, setting configuration parameters.
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: user help articles http://id.ee/index.php?id=30591
2. PIN dialog – for inserting PIN value during signature creation or decryption operations in all operating systems except of Windows
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: -

Required:

1. Kill switch interface: see chap. 3.6.2
2. DigiDocService web service interface: see chap. 3.6.3
3. Error reports repository interface: see chap. 3.6.4
4. LDAP directory interface: see chap. 3.6.5
5. Interfaces with base libraries:
    a. Libdigidocpp library's API – for handling documents in supported digital signature formats (BDOC and DDOC). See chap. 0
    b. Other base libraries: see DigiDoc3 Client application's documentation (to be included in the distribution package).
6. Interfaces with cryptographic token's drivers:
    a. PKCS#11 interface: see chap. 3.4.1
    b. CNG interface: see chap. 3.4.1

3.1.2 ID-card utility

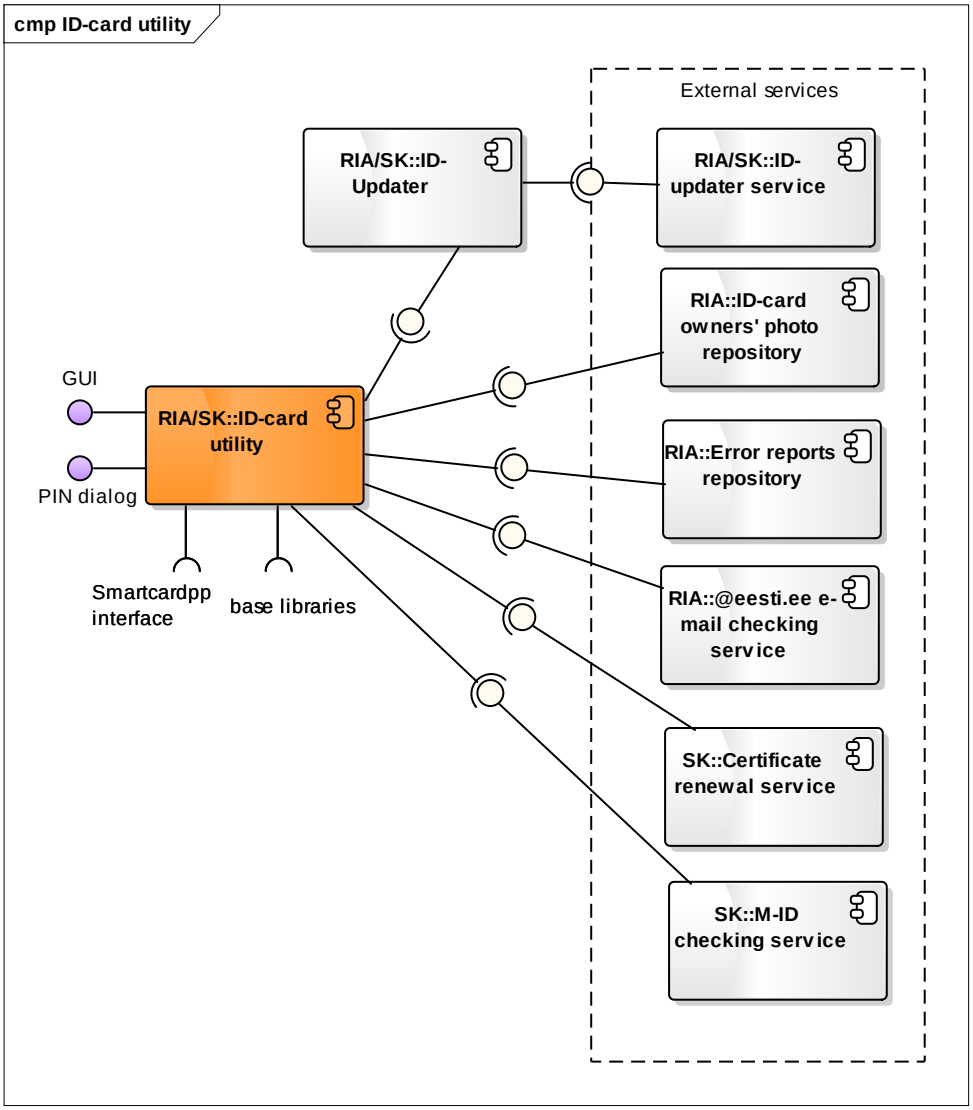

Figure 2. Components of ID-card utility

Table 4. Components of ID-card utility

Component	Description	Owner/ Developer
ID-card utility	End-user desktop application for managing ID-card's PIN/PUK codes replacement, certificates' renewal and other services.	RIA/SK
ID-card utility's base libraries	See ID-card utility's documentation (to be included in the distribution package).	-
ID-card owner's photo repository	Repository where the Estonian national ID-cards photos' are kept. ID-card's owner can download the photo after the user has been authenticated with PIN1 code.	RIA
Error reports repository	Described in chap. 3.1.1	RIA
@eesti.ee e-mail checking service	Service that enables to set the properties of e-mail address (@eesti.ee) that is provided for Estonian national ID-card owners by the state. The user must be authenticated with PIN1 code.	RIA

Component	Description	Owner/ Developer
Certificate renewal service	Service for renewing certificates on the Estonian national ID-card.	SK
M-ID checking service	Service for checking the status of Estonian national ID-card owner's Mobile-ID certificates. The user must be authenticated with PIN1 code.	SK
Updater, ID-updater Service	Described in chap. 3.4.1	RIA/SK

3.1.2.1 Interfaces

Provided:

1. Graphical user interface – interface for handling card management operations and using the external services (listed under "Required interfaces").
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: ID-card utility program's documentation (to be included in the distribution package).
2. PIN dialog – for inserting PIN/PUK value in all supported operating systems.
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: -

Required:

1. ID-card owner photos' repository: see chap. 3.6.9
2. Error reports repository interface: see chap. 3.6.4
3. Certificate renewal service interface: see chap. 3.6.10
4. Eesti.ee e-mail checking service interface: see chap: 3.6.11
5. Certificate renewal service, see chap. 3.6.10
6. Mobile-ID validity checking service interface: see chap. 3.6.12
7. Interfaces of base libraries: see ID-card utility program's documentation (to be included in the distribution package).
8. Interfaces with cryptographic token's drivers:
    a. Smartcardpp API (internal component)

3.2 Software libraries

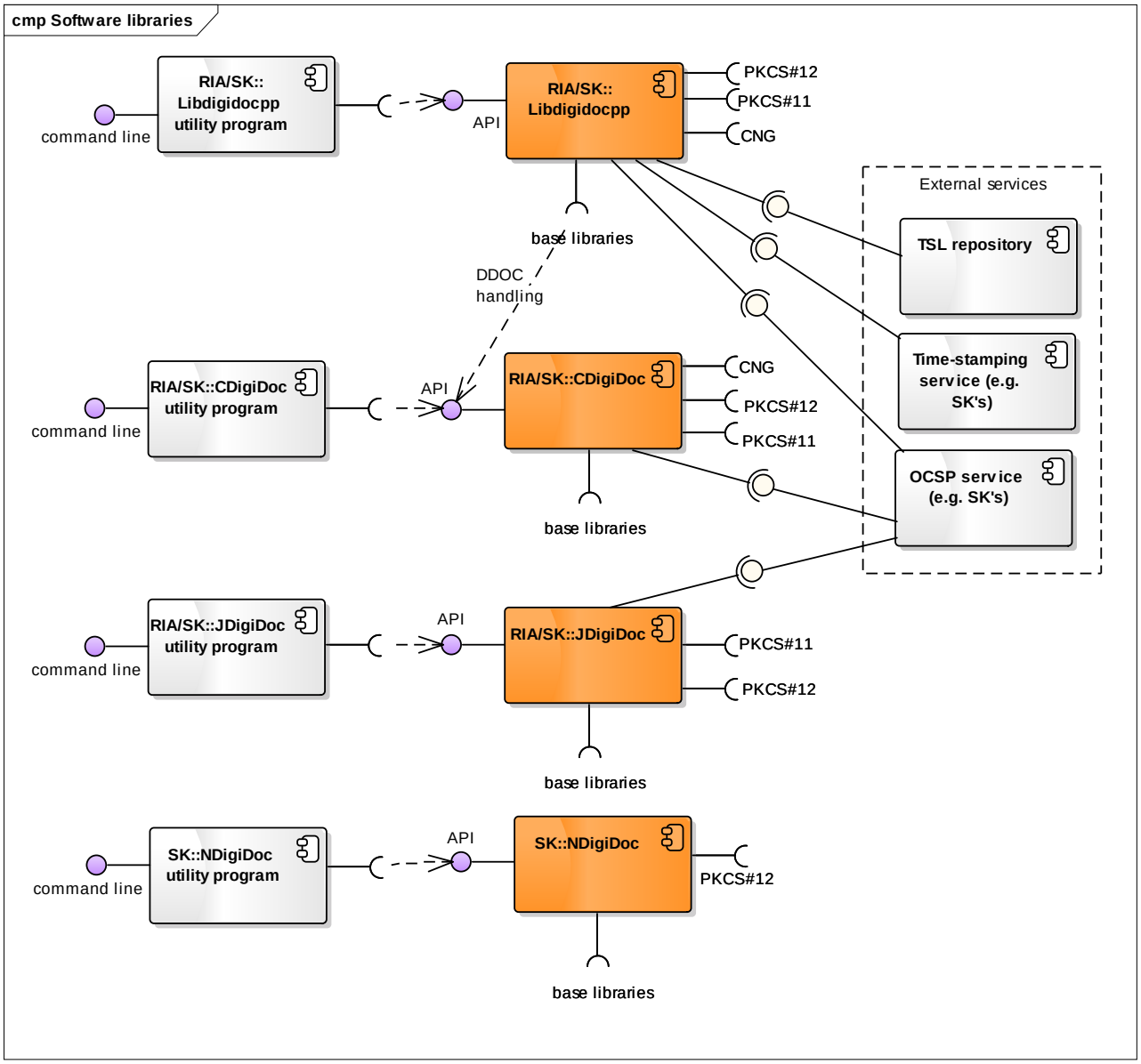

Figure 3. Software libraries and their components

Table 5. Software libraries and their components

Component	Description	Owner/ Developer
Libdigidocpp	C++ software library that enables handling documents in BDOC 2.1 [1] and DIGIDOC-XML 1.3 [3] formats (via CDigiDoc base library). See [7] for additional information about the library and its base components.	RIA/SK
Libdigidocpp utility program	Small command line application (digidoc-tool.exe) that implements the main functionality of Libdigidocpp library. Used for testing purposes. Can also be used as a source for sample client code for using Libdigidocpp. See [7] for additional information.	RIA/SK
CDigiDoc	Software library in C that enables handling digitally signed documents in DIGIDOC-XML 1.3 [3] format and encryption/decryption in ENCDOC-XML 1.0 (CDOC) [5]. See [9] for additional information about the library and its base components.	RIA/SK

Component	Description	Owner/ Developer
CDigiDoc utility program	Small command line application that implements the main functionality of CDigiDoc library. Used for testing purposes. Can also be used as a source for sample client code for using CDigiDoc. See [9] for additional information.	RIA/SK
JDigiDoc	Java software library that enables handling documents in BDOC 2.1 [1] and DIGIDOC-XML 1.3 [3] formats and encryption/decryption in ENCDOC-XML 1.0 (CDOC) [5]. See [8] for additional information about the library and its base components.	RIA/SK
JDigiDoc utility program	Small command line application that implements the main functionality of JDigiDoc library. Used for testing purposes. Can also be used as a source for sample client code for using JDigiDoc. See [8] for additional information.	RIA/SK
NDigiDoc	Software library in .NET enabling encryption/decryption in ENCDOC-XML 1.0 (CDOC) [5]. See [10] for additional information about the library and its base components.	SK/SK
NDigiDoc utility program	Small command line application that implements the main functionality of NDigiDoc library. Used for testing purposes. Can also be used as a source for sample client code for using NDigiDoc. See [10] for additional information.	SK/SK
TSL repository	Repository for accessing the TSL (Trust Service status List, [15]) lists that can be used as a central source of trust anchor information during digital signature creation and validation processes. The European Commission's TSL list (https://ec.europa.eu/information_society/policy/esignature/trusted-list/tl-mp.xml) is used as the master list. See also [7] for additional information on TSL implementation (currently only in Libdigidocpp library).	-
Time-stamping service	Described in chap. 3.1.1	-
OCSP service	Described in chap. 3.1.1	-

3.2.1 Libdigidocpp library's interfaces

Provided:

1. Libdigidocpp API
    a. User: server application, end-user application
    b. Accessible with: C++
    c. Documentation: see [7]

Required:

1. TSL repository interface: see chap. 3.6.6
2. Time-stamping service interface: see chap. 3.6.7
3. OCSP service interface: see chap. 3.6.8
4. Interfaces with base libraries:
    a. CDigiDoc library's API – for handling documents in DDOC format ([3]). See chap. 3.2.5.
    b. Other base libraries: see [7].
5. Interfaces with cryptographic token's drivers:
    a. PKCS#11 interface: see chap. 3.4.1
    b. CNG interface: see chap. 3.4.1
    c. PKCS#12 interface: see chap. 3.4.1

3.2.2 Libdigidocpp utility program's interfaces

Provided:

1. Libdigidocpp utility program's interface
    a. User: server application, end-user application, end-user
    b. Accessible with: command line
    c. Documentation: see [7]

Required:

1. Libdigidocpp API: see chap. 0

3.2.3 JDigiDoc library's interfaces

Provided:

1. JDigiDoc API
    a. User: server application, end-user application
    b. Accessible with: Java
    c. Documentation: see [8]

Required:

1. OCSP service interface: see chap. 3.6.8
2. Interfaces with base libraries: see [8] for more information.
3. Interfaces with cryptographic token's drivers:

a. PKCS#11 interface: see chap. 3.4.1
b. CNG interface: see chap. 3.4.1
c. PKCS#12 interface: see chap. 3.4.1

3.2.4 JDigiDoc utility program's interfaces

Provided:

1. JDigiDoc utility program's interface
    a. User: server application, end-user application, end-user
    b. Accessible with: command line
    c. Documentation: see [8]

Required:

1. JDigiDoc API: see chap. 3.2.3

3.2.5 CDigiDoc library's interfaces

Provided:

1. CDigiDoc API
    a. User: server application, end-user application
    b. Accessible with: C
    c. Documentation: see [9]

Required:

1. OCSP service interface: see chap. 3.6.8
2. Interfaces with base libraries: see [9].
3. Interfaces with cryptographic token's drivers:
    a. PKCS#11 interface: see chap. 3.4.1
    b. CNG interface: see chap. 3.4.1
    c. PKCS#12 interface: see chap. 3.4.1

3.2.6 CDigiDoc utility program's interfaces

Provided:

1. CDigiDoc utility program's interface
    a. User: server application, end-user application, end-user
    b. Accessible with: command line/console
    c. Documentation: see [9]

Required:

1. CDigiDoc API: see chap. 3.2.5

3.2.7 NDigiDoc library's interfaces

Provided:

1. NDigiDoc API
    a. User: server application, end-user application
    b. Accessible with: .NET
    c. Documentation: see [10]

Required:

1. Interfaces with base libraries: see [10] for more information.
2. Interfaces with cryptographic token's drivers:
    a. PKCS#12 interface: see chap. 3.4.1

3.2.8 NDigiDoc utility program's interfaces

Provided:

1. NDigiDoc utility program's interface
    a. User: server application, end-user application, end-user
    b. Accessible with: command line/console
    c. Documentation: see [10]

Required:

1. NDigiDoc API: see chap. 3.2.7

3.3 Web components

3.3.1 Web signing components

The web signing component diagrams describe components that are needed for signature creation in web applications with eID cards.

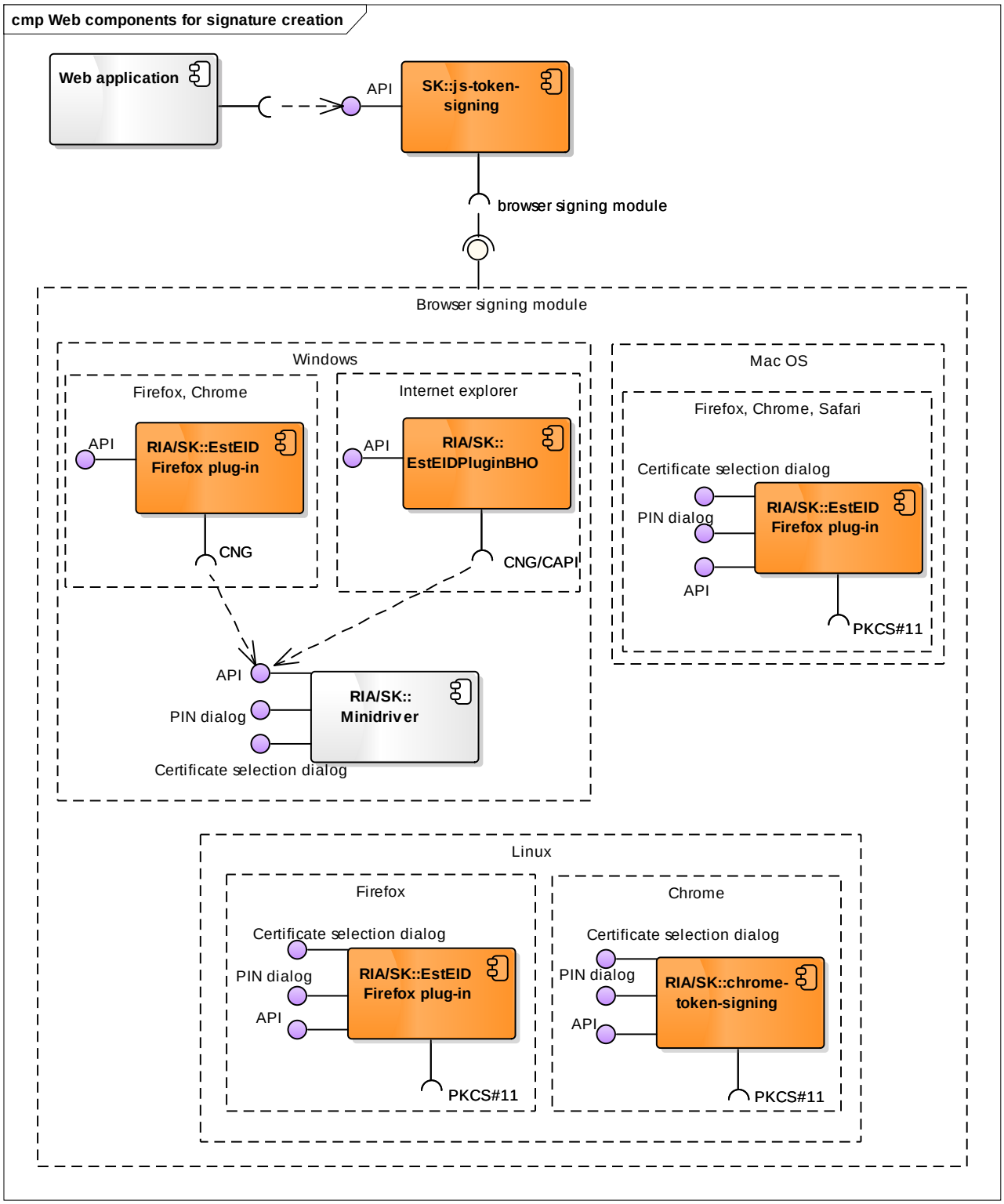

Table 6. Components for signing in web environment

Component	Description	Owner/ Developer
js-token-signing	JavaScript library (also known as idCard.js) that enables communication with the browser signing module (plug-in or extension). See also [14] for more information.	SK/SK
Web application	A web application that implements signature creation with an eID-card in browser environment.	-
EstEID Firefox plug-in	Browser signing module (NPAPI-based plug-in) that is used in Firefox, Chrome and Safari browsers in all of the supported operating systems. The plug-in enables data exchange with the cryptographic token's driver that is used for signing.	RIA/SK
EstEIDPluginBHO	Browser signing module (BHO-based plug-in) that is used in Internet explorer browser in Windows operating system. The plug-in enables data exchange with the cryptographic token's driver that is used for signing.	RIA/SK
chrome-token-signing	Browser signing module (native messaging extension) that is used in Chrome browser in Linux operating system. The extension enables data exchange with the cryptographic token's driver that is used for signing	RIA/SK
Minidriver	Used via CNG interface in Windows environment only. Described in chap. 3.4	RIA/SK

3.3.1.1 Js-token-signing library's interfaces

Provided:

1. Js-token-signing library's API
    a. User: a web application in browser environment
    b. Accessible with: JavaScript
    c. Documentation: see [14]

Required:

1. Interfaces with browser signing modules:
    a. EstEID FireFox plug-in API: see chap. 3.3.1.2
    b. EstEIDPluginBHO plug-in API: see chap. 3.3.1.3
    c. Chrome-token-signing extension API: see chap. 3.3.1.4

3.3.1.2 EstEID Firefox plug-in's interfaces

Provided:

1. EstEID Firefox plug-in's API
    a. User: a web application in browser environment, js-token-signing library
    b. Accessible with: C
    c. Documentation: see [14]
2. PIN dialog – for inserting PIN2 value during signature creation in all operating systems except of Windows
    a. User: end-user
    b. Accessible with: GUI elements

c. Documentation: -

3. Certificate selection dialog
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: -

Required:

1. Interfaces with cryptographic token's drivers:
    a. PKCS#11 interface: see chap. 3.4.1
    b. CNG interface: see chap. 3.4.1

3.3.1.3 EstEIDPluginBHO plug-in's interfaces

Provided:

1. EstEIDPluginBHO plug-in's API
    a. User: a web application in browser environment, js-token-signing library
    b. Accessible with: C++
    c. Documentation: see [14]

Required:

1. Interfaces with cryptographic token's drivers:
    a. CNG/CAPI interface: see chap. 3.4.1

3.3.1.4 Chrome-token-signing extension's interfaces

Provided:

1. Chrome-token-signing extension's API
    a. User: a web application in browser environment, js-token-signing library
    b. Accessible with: C++
    c. Documentation: see [14]
2. PIN dialog – for inserting PIN2 value during signature creation
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: -
3. Certificate selection dialog
    a. User: end-user
    b. Accessible with: GUI elements
    c. Documentation: -

Required:

1. Interfaces with cryptographic token's drivers:
    a. PKCS#11 interface: see chap. 3.4.1

3.3.2 Web authentication components

Authentication in web browsers is done with the browsers' and operating systems' native components. In case of authenticating in Firefox browser then Firefox pkcs11-module-loader JavaScript component is used to load the OpenSC PKCS#11 driver by the browser.

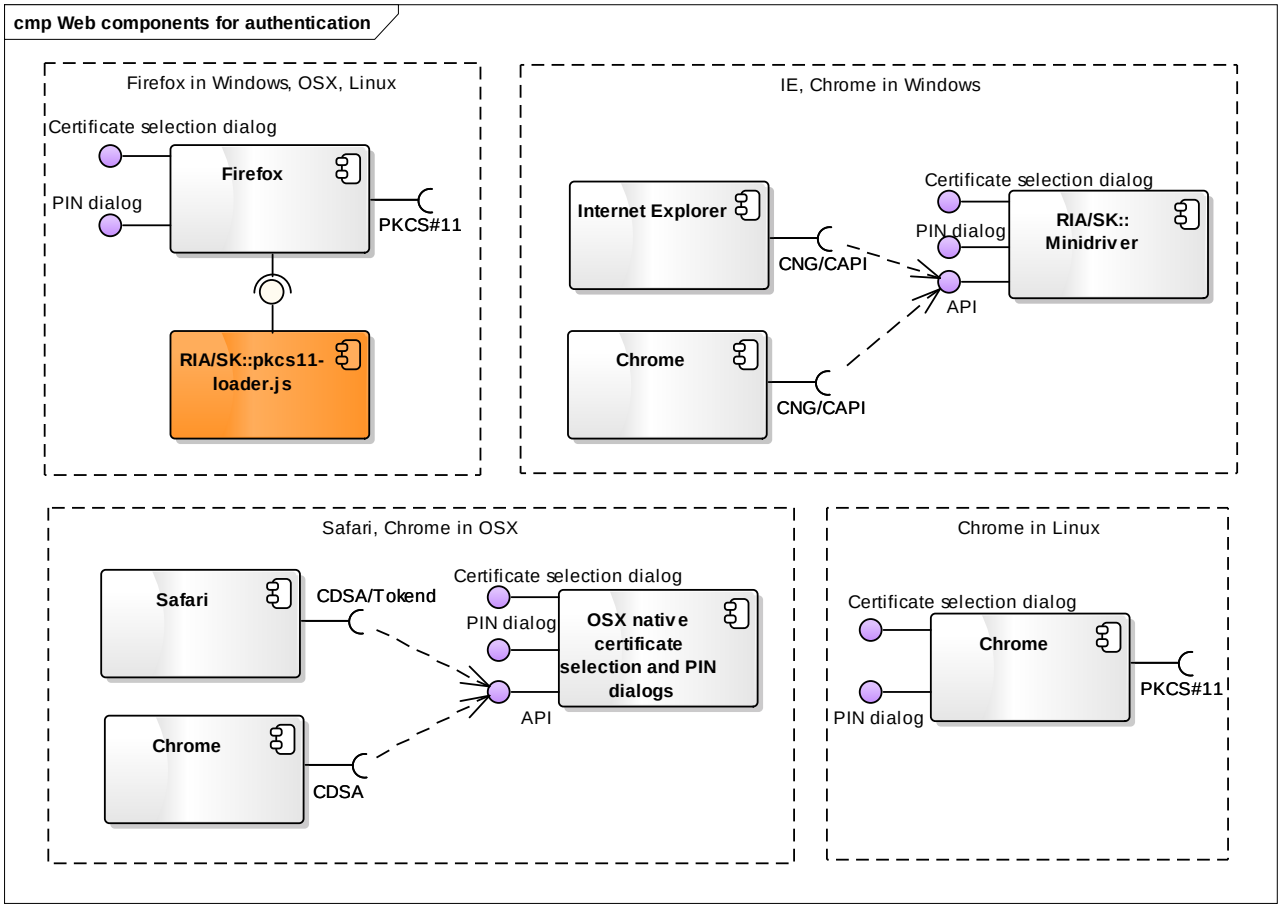

Figure 4. Web authentication components

Table 7. Web authentication components

Component	Description	Owner/ Developer
pkcs11-loader.js	A JavaScript component that is used to load the OpenSC PKCS#11 driver to the Firefox browser's cryptographic devices list during each initialization of the browser. Needed during authentication process with eID-card in Firefox browser in all supported operating systems.	RIA/SK
OSX native certificate selection and PIN dialog	PIN dialog and certificate selection windows provided by the operating system's native components.	-

3.4 Drivers

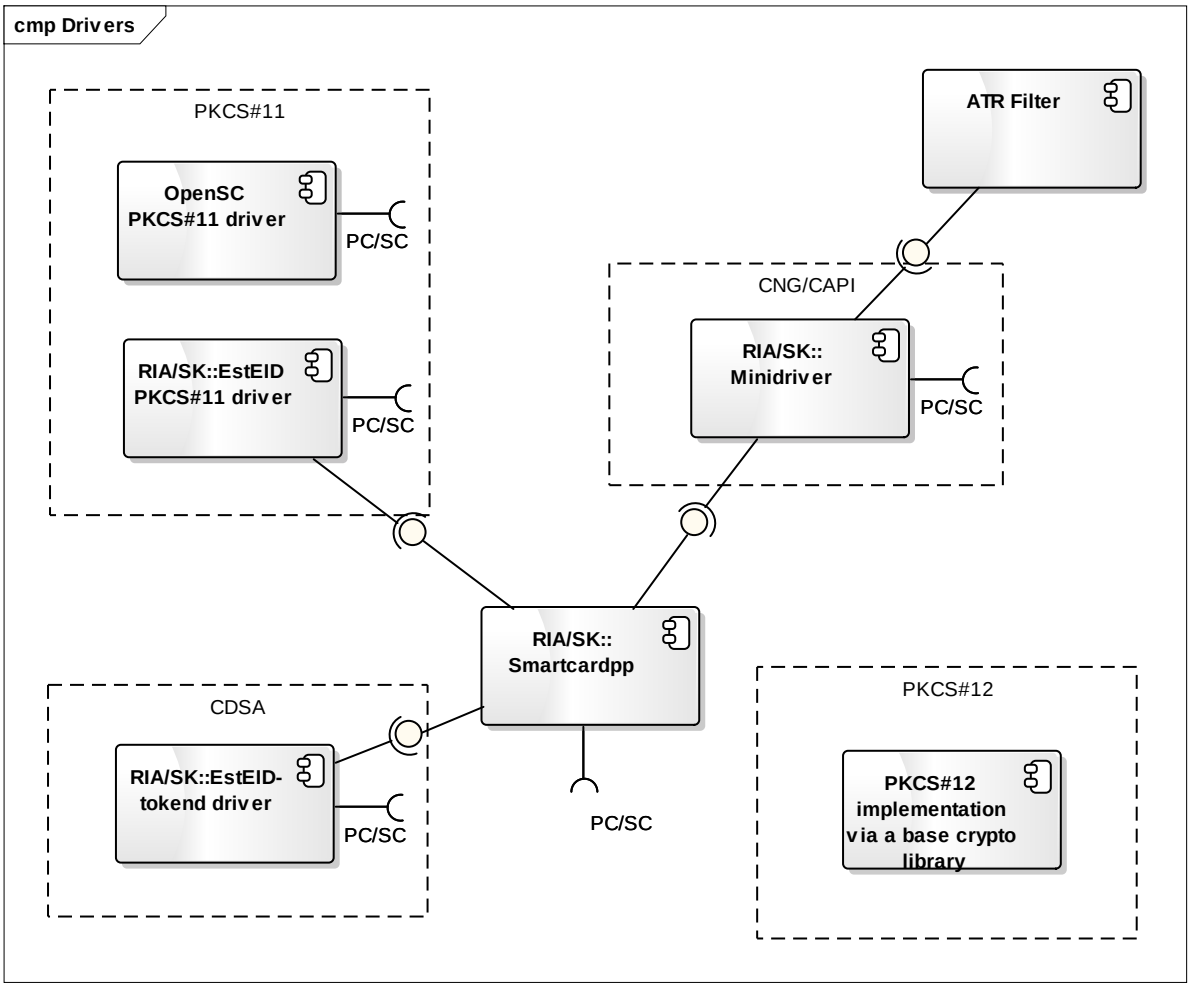

Figure 5. Cryptographic tokens' drivers

Table 8. Web authentication components

Component	Description	Owner/ Developer
EstEID PKCS#11 driver	A driver for accessing eID-cards. Connects with the card via the operating system's native PC/SC interface. Used as a default driver for signature creation with eID card in browser environment in case of OSX platform. Used as a default driver for authentication with eID card in browser environment in case of Firefox browser in OSX platform.	RIA/SK
OpenSC PKCS#11 driver	A driver for accessing eID-cards. Connects with the card via the operating system's native PC/SC interface. Used as a default driver for authentication with eID card in browser environment in case of Windows and Linux platforms. Used as a default driver for signature creation in web browser environment in case of Linux platform.	-
Smartcardpp	eID card driver's helper component. Inner component.	RIA/SK

Component	Description	Owner/ Developer
Minidriver	Used as a default driver for accessing eID-cards via CNG interface for signature creation in web browser environment in case of Windows platform. Used as a default driver for authentication with eID card in Chrome and Internet Explorer browsers in case of Windows platform.	RIA/SK
ATR Filter	Base component for Minidriver (see http://support.microsoft.com/kb/981665 for more information).	-
Esteid Tokend	A driver for accessing eID-cards. Connects with the card via the operating system's native PC/SC interface. Used as a default driver for authentication with eID card in browser environment in case OSX platform.	RIA/SK
PKCS#12 implementation via base library	An implementation of PKCS#12 interface by the component's base libraries.	-

3.4.1 Cryptographic tokens drivers' interfaces

3.4.1.1 PKCS#11 drivers

Components:

1. EstEID PKCS#11 driver
2. OpenSC PKCS#11 driver

Provided:

1. PKCS#11 API
    a. User: browser signing module, software library
    b. Accessible with: C++
    c. Documentation:
        i. PKCS#11 API: http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-11-cryptographic-token-interface-standard.htm
        ii. source code for the list of implemented functions

Required:

1. PC/SC: see chap. 3.4.1.5

3.4.1.2 Minidriver

Provided:

1. CNG/Minidriver API
    a. User: browser signing module, software library
    b. Accessible with: C/C++
    c. Documentation:
        i. CNG: http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx,
        ii. Minidriver API: http://msdn.microsoft.com/en-us/library/windows/hardware/dn631754(v=vs.85).aspx

iii. source code for the list of implemented functions

2. CAPI/Minidriver API
    a. User: browser signing module, software library
    b. Accessible with: C/C++
    c. Documentation:
        i. CAPI: http://msdn.microsoft.com/en-us/library/aa380256.aspx
        ii. Minidriver API: http://msdn.microsoft.com/en-us/library/windows/hardware/dn631754(v=vs.85).aspx
        iii. source code for the list of implemented functions

Required:

1. PC/SC: see chap. 3.4.1.5

3.4.1.3 PKCS#12 implementation via base library

Provided:

1. PKCS#12 interface
    a. User: software library
    b. Accessible with: PKCS#12 API
    c. Documentation: see documentation of the respective components appropriate base library

3.4.1.4 Tokend driver

Components implementing the interface:

1. EstEID Tokend driver

Provided:

1. CDSA
    a. User: software library
    b. Accessible with: C++
    c. Documentation: see https://developer.apple.com/library/mac/documentation/security/conceptual/cryptoservices/CDSA/CDSA.html

Required:

1. PC/SC: see chap. 3.4.1.5

3.4.1.5 PC/SC driver

Provided:

1. PC/SC interface
    a. User: eID-card's driver
    b. Accessible with: PC/SC API

c. Documentation: see http://www.pcscworkgroup.com/specifications/overview.php

Required: not in the scope of this document.

3.5 Updating mechanisms

The following chapter describes automatic updating mechanisms of different ID-software desktop applications. Several combinations of central software update checking and distribution environments are used depending on the end-user's operating system.

3.5.1 Windows

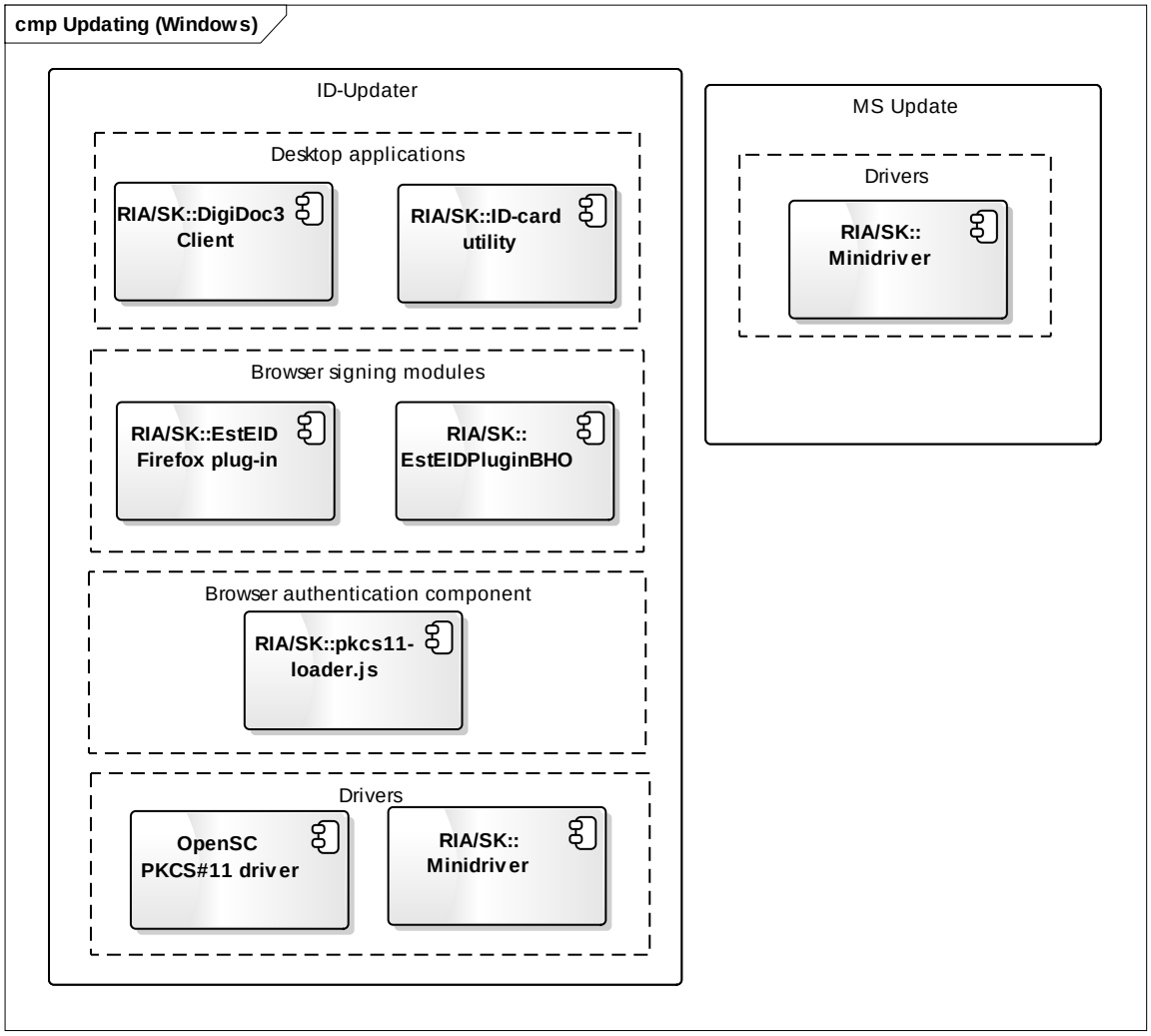

Figure 6. Updating mechanisms in Windows

Table 9. Updating mechanisms in Windows

Component	Description	Owner/ Developer
ID-updater	Service that is periodically checks if newer versions of related ID-software components are available for download, initiates the download and installation if necessary.	RIA/SK
MS Updater	Microsoft Updater – see Microsoft's documentation for more information.	-

3.5.2 OS X

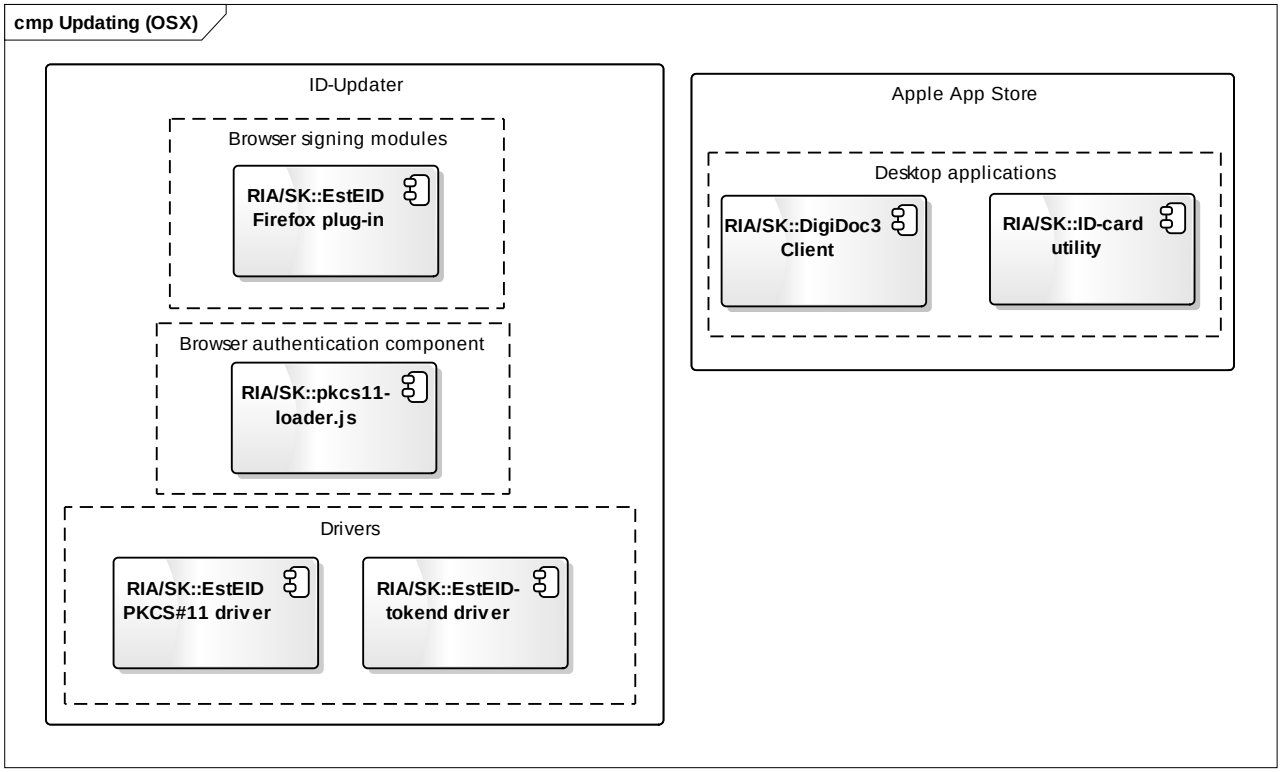

Figure 7. Updating mechanisms in OSX

Table 10. Updating mechanisms in OSX

Component	Description	Owner/ Developer
ID-updater	See chap. 3.6.1	RIA/SK
Apple App Store	See Apple App Store documentation.	-

3.5.3 Linux

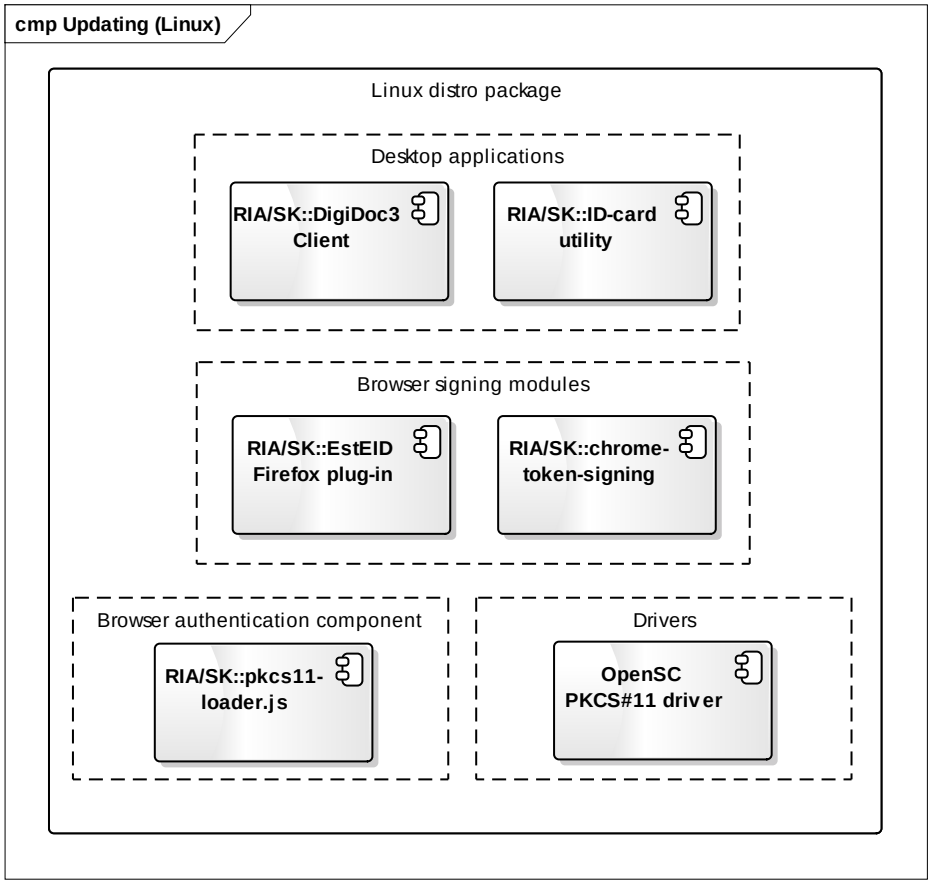

Figure 8. Updating mechanism in Linux

Table 11. Updating mechanisms in OSX

Component	Description	Owner/ Developer
Ubuntu package updates	Managed and maintained by SK. The binary packages are released for installation and updating to https://installer.id.ee/media/ubuntu/ repository.	RIA/SK
Packages updates for other distros	Managed by the open-source community. Packages are built, added and updated in Estobuntu and Fedora distributions by the package maintainers.	-

3.6 External services' interfaces

3.6.1 ID-updater interface

Interface's properties:

1. User: DigiDoc3 Client desktop application
2. Accessible with: HTTPS protocol
3. Accessible from:
    a. Windows: https://installer.id.ee/media/win/products.xml,
    b. OSX: https://installer.id.ee/media/osx/products.xml,
    c. Linux: https://installer.id.ee/media/ubuntu/pool/main/
4. Documentation: -

3.6.2 Kill switch service interface

Interface's properties:

1. User: DigiDoc3 Client/DigiDoc3 Crypto desktop application
2. Accessible with: XML file sent over HTTPS protocol
3. Accessible from: https://installer.id.ee/media/killswitch/products.xml
4. Documentation: -

3.6.3 DigiDocService web service interface

Interface's properties:

1. User: DigiDoc3 Client desktop application
2. Accessible with: SOAP 1.0-encoded over HTTPS
3. Accessible from: https://digidocservice.sk.ee
4. Documentation: see [12]

3.6.4 Error reports repository interface

Interface's properties:

1. User: DigiDoc3 Client/Crypto, ID-utility desktop applications
2. Accessible with: HTTPS protocol
3. Accessible from: https://cr.eesti.ee
4. Documentation: -

3.6.5 LDAP directory interface

Interface's properties:

1. User: DigiDoc3 Crypto desktop application

2. Accessible with: LDAP protocol
3. Accessible from: ldap.sk.ee:389
4. Documentation: [13]

3.6.6 TSL repositories' interfaces

Interface's properties:

1. User: Libdigidocpp software library
2. Accessible with: HTTPS protocol
3. Accessible from:
    a. European Commission's master list: https://ec.europa.eu/information_society/policy/esignature/trusted-list/tl-mp.xml
    b. Finnish TSL: https://www.viestintavirasto.fi/attachments/TSL-Ficora.xml
    c. Estonian TSL: http://sr.riik.ee/tsl/estonian-tsl.xml
    d. Latvian TSL: http://www.mca.org.mt/tsl/MT_TSL.xml
    e. Lithuanian TSL: http://www.rrt.lt/failai/LT-TSL.xml
4. Documentation: see [15]

3.6.7 Time-stamping service interface

Interface's properties:

1. User: Libdigidocpp software library
2. Accessible with: HTTPS protocol
3. Accessible from:
    a. SK's time-stamping service http://demo.sk.ee/tsa/
4. Documentation: see RFC3161

3.6.8 OCSP service interface

Interface's properties:

1. User: JDigiDoc, Libdigidocpp CDigiDoc software libraries; DigiDocService web service
2. Accessible with: HTTPS protocol
3. Accessible from:
    a. SK's OCSP service for SK issued certificates: http://ocsp.sk.ee/
    b. SK's Proxy OCSP service for international use: http://ocsp.sk.ee/_proxy
4. Documentation: see RFC2560, RFC6960

3.6.9 ID-card owners' photo repository interface

Interface's properties:

1. User: ID-card utility program
2. Accessible with: HTTPS protocol
3. Accessible from: https://sisene.www.eesti.ee/idportaal/portaal.idpilt
4. Documentation: -

3.6.10 Certificate renewal service interface

Interface's properties:

1. User: ID-card utility program
2. Accessible with: HTTP
3. Accessible from: http://www.sk.ee:80/id-kontroll2/usk/
4. Documentation: -

3.6.11 Eesti.ee e-mail checking service interface

Interface's properties:

1. User: ID-card utility program
2. Accessible with: HTTPS
3. Accessible from: https://sisene.www.eesti.ee/idportaal/postisysteem.naita_suunamised
4. Documentation: -

3.6.12 Mobile-ID validity checking service interface

Interface's properties:

1. User: ID-card utility program
2. Accessible with: HTTPS
3. Accessible from: https://id.sk.ee/MIDInfoWS/
4. Documentation: -

4 Deployment model

The following subchapters describe physical deployment of ID-software components in collaboration with external components that were depicted in chap. “3 Component model” in case of the most common use cases.

4.1 Signing in web browser

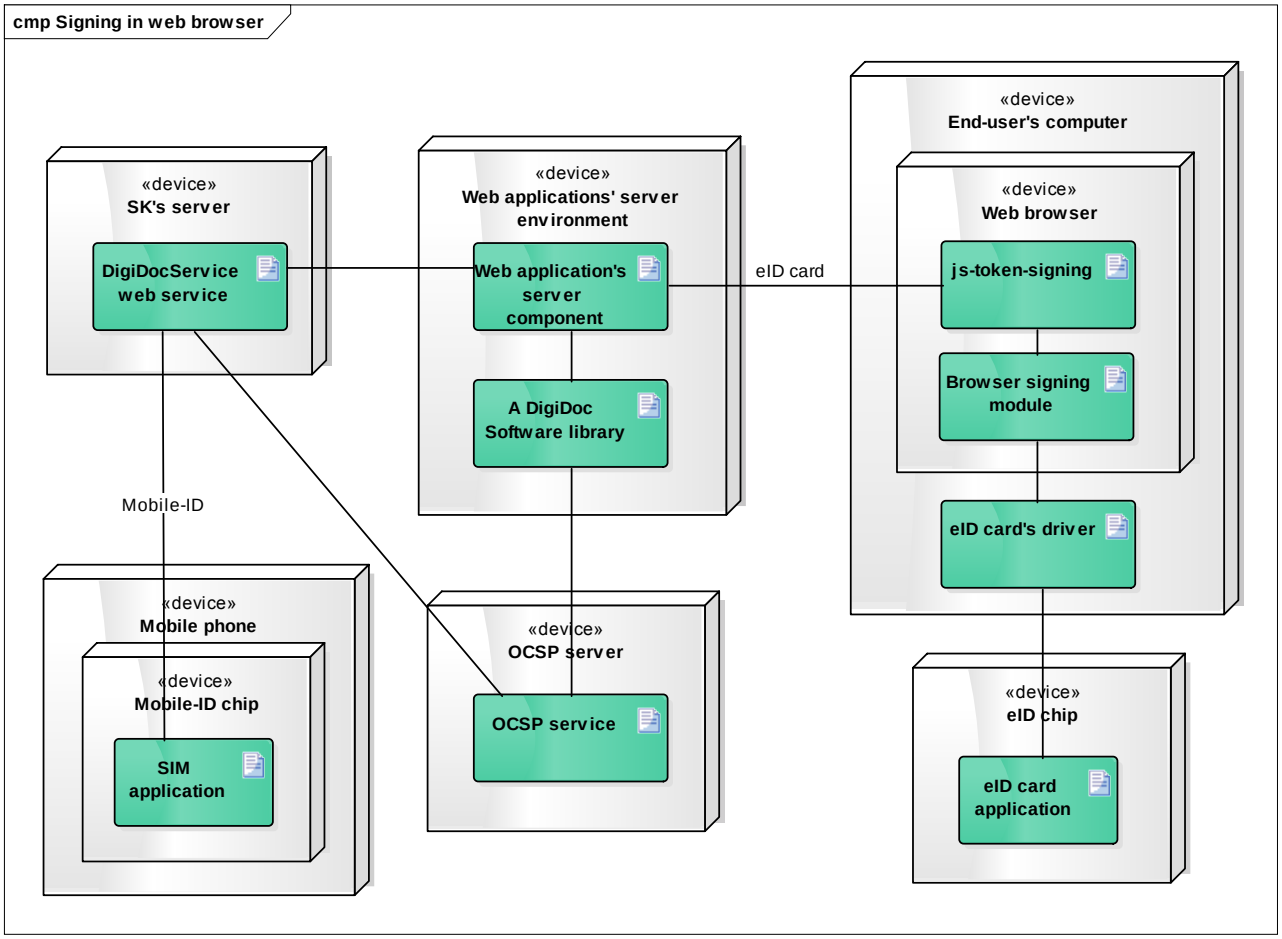

Figure 9. Signing in web browser via a web application

Additional notes:

- A DigiDoc software library (i.e. JDigiDoc, Libdigidocpp or CDigiDoc library) and DigiDocService web service are optional and can be used for adding the created signature value to a DDOC or BDOC container.
- OCSP service is required when creating qualified DDOC/BDOC signatures.
- DigiDocService is required in order to sign with Mobile-ID.
- Signature value is calculated either in the Mobile-ID SIM card or eID-card’s chip.

4.2 Signing with DigiDoc3 Client

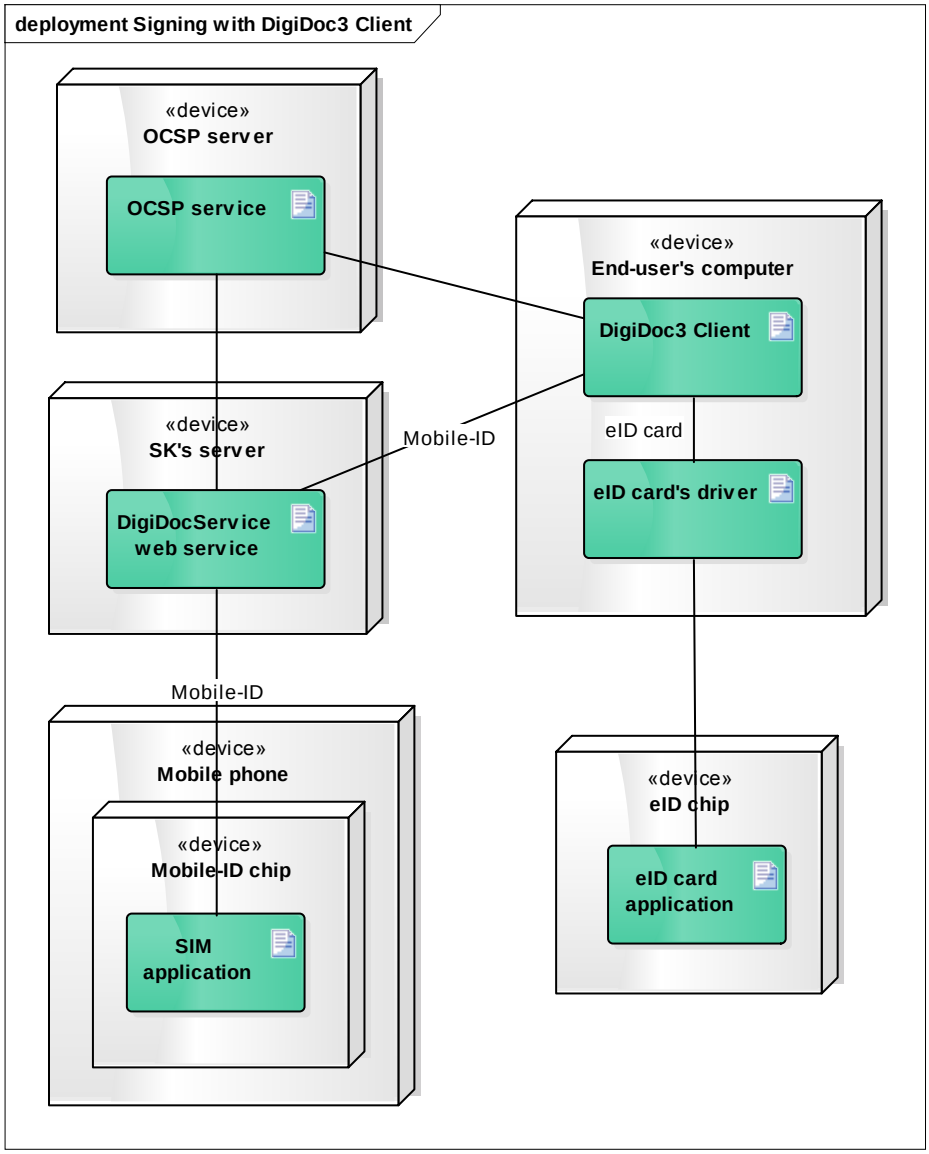

Figure 10. Deployment of components during signature creation with DigiDoc3 Client

Additional notes:

- The DigiDoc3 Client is used for adding the created signature value to a DDOC or BDOC container.
- DigiDocService is required in order to sign with Mobile-ID.
- Signature value is calculated either in the Mobile-ID SIM card or ID-card's chip.

5 References

[1] BDOC – format for digital signatures: OID 1.3.6.1.4.1.10015.1000.3.2.3. Version 2.1.2:2014 [WWW] http://id.ee/public/bdoc-spec212-eng.pdf

[2] BDOC file format, what is it, when will it replace DDOC format and what's needed for transition? [WWW] http://www.id.ee/?lang=en&id=34336

[3] DigiDoc Format Specification. Version 1.3.0 [WWW] http://id.ee/public/DigiDoc_format_1.3.pdf

[4] ETSI TS 101 903 V1.4.2 (2010-12) – XML Advanced Electronic Signatures [WWW] http://www.etsi.org/deliver/etsi_ts/101900_101999/101903/01.04.02_60/ts_101903v010402p.pdf

[5] Encrypted DigiDoc Format Specification. Version 1.1 [WWW] http://id.ee/public/SK-CDOC-1.0-20120625_EN.pdf

[6] List of latest ID-software component versions. [WWW] http://www.id.ee/?lang=en&id=36798

[7] Libdigidocpp Programmer's Guide. [WWW] /TBS/

[8] JDigiDoc Programmer's Guide. [WWW] http://id.ee/public/SK-JDD-PRG-GUIDE.pdf

[9] CDigiDoc Programmer's Guide. [WWW] http://id.ee/index.php?id=35782

[10] NDigiDoc Programmer's Guide. [WWW] http://id.ee/public/NDigiDoc.pdf

[11] List of URL's to where ID-card software internally connects. [WWW] http://id.ee/index.php?id=35904

[12] DigiDocService specification. [WWW] http://www.sk.ee/upload/files/DigiDocService_spec_eng.pdf

[13] LDAP technical description. [WWW] https://sk.ee/en/repository/ldap/ldap-kataloogi-kasutamine/

[14] JavaScript Library for Signing in Web Browsers. [WWW] http://id.ee/public/web_sign_library.zip

[15] ETSI TS 119 612 V1.2.1 (2014-04) - Trusted Lists [WWW] http://www.etsi.org/deliver/etsi_ts/119600_119699/119612/01.02.01_60/ts_119612v010201p.pdf