Implementation Guide for LSS

Table of contents

Version history

31th January 2014	Version 0.92	MSP
10th January 2014	Version 0.91	MSP
20th December 2013	Version 0.6	MSP
16th December 2013	Version 0.5	MSP

1 The purpose and audience of the document

The purpose of this document is to provide implementation guidelines for implementing support for the LSS for NemID API in Central Signature servers products.

The document is aimed at the developers at the LSS supplier organization who are responsible for developing support of the LSS for NemID API in the the relevant LSS product.

Summary of all documents in the LSS for NemID Package:

Implementation documentation

- Technical specification for LSS
- Implementation guide for LSS

Test documentation

- Guidelines on the use of LSS for NemID test tools
- Testprocedures for LSS

Solution documentation

- Requirements feedback form for LSS
- End-customer documentation for LSS

2 Introduction to LSS for NemID

For a general introduction to NemID and NemID for business consult the current service provider package (TU-pakke) from Nets DanID[1].

For the rest of this document, knowledge of the general concept of NemID and NemID for business, as found in the current service provider package, is expected.

As a supplier of LSS products it is possible to integrate to the LSS for NemID. This makes it possible for employees at companies with the LSS to use NemID for business from JavaScript enabled devices such as tablets, smartphones and ordinary computers with no need to use specific plug-ins. Note that service providers may and may not choose to support the LSS for NemID functionality in their services. The purpose of the NemID service provider package for Local Signature servers (LSS for NemID) is to provide a JavaScript based integration between service providers (SP) and employees at organizations, who have their NemID for business stored on a local signature server hosted on their enterprise LAN.

A test/reference implementation based on .Net of a LSS-backend integrating with the LSS for NemID setup is available in the package. LSS suppliers can choose to develop their own implementation in other technologies, if they wish to do so, as long as the technical specifications for the defined LSS for NemID protocol and API towards services providers are respected - as defined in the document "Technical specification for LSS".

This document walks through the requirements to the LSS supplier's implementation of LSS for NemID.

[1] https://www.nets-danid.dk/tu-pakke

3 Solution architecture

The purpose of the LSS for NemID is to provide the ability of employees of organizations with a LSS to authenticate towards service providers and to digitally sign documents in formats Text, HTML, XML and PDF.

To support this functionality, the LSS supplier needs to implement authentication and signing capabilities towards their own LSS backend.

The overall architecture is illustrated below.

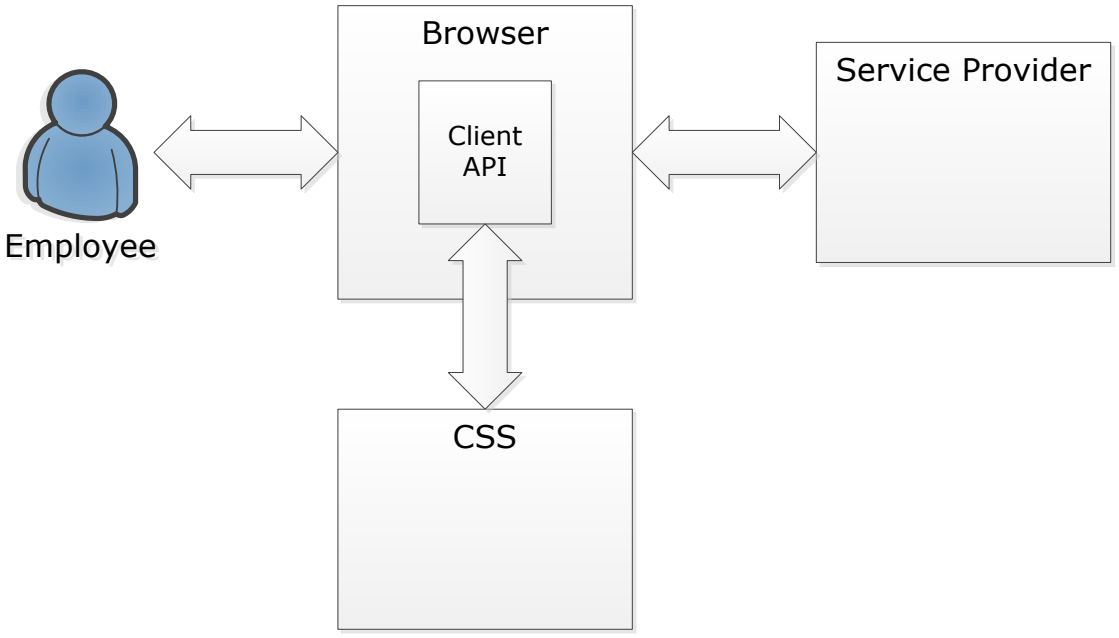

Fig 1: Overall architecture

3.1 Responsibilities of Service providers and LSS suppliers

The responsibility and role of the service provider is exactly the same as with the existing NemID solutions.

The responsibility and role of the LSS supplier is to implement all the functionality required by the service provider in order to provide authentication of the user and on success return a XML-DSig message to the service provider.

1. The service provider either sends a challenge to be signed for authentication or a document to be signed.

2. The challenge or the document to be signed is then presented to the user by the LSS suppliers implementation of the content of the iFrame (UX).

3. The user authenticates towards the LSS, and the challenge or the document is signed and returned to the service provider in XML-DSig format.

4. The service provider verifies the returned XML-DSig with the existing OOAPI tool of the DanID TU software package, following the exact same procedures for trust, revocation check of certificates, validity check etc., as when validating XML-DSig received from other NemID variants.

It is thus the responsibility of the LSS supplier's implementation to present to the user the text or document to be signed. This implementation must support "What you see is what you sign" and the further limitations on the content when presented as HTML or PDF as specified in the document "Technical specification for LSS".

3.2 IFrame integration

The JavaScript LSS for NemID client is integrated with the service provider's page using an <iFrame> element, which enables a web page to allocate a segment of its area to another page. This is a change from the Java applet client, where a Java applet was loaded as a page element.

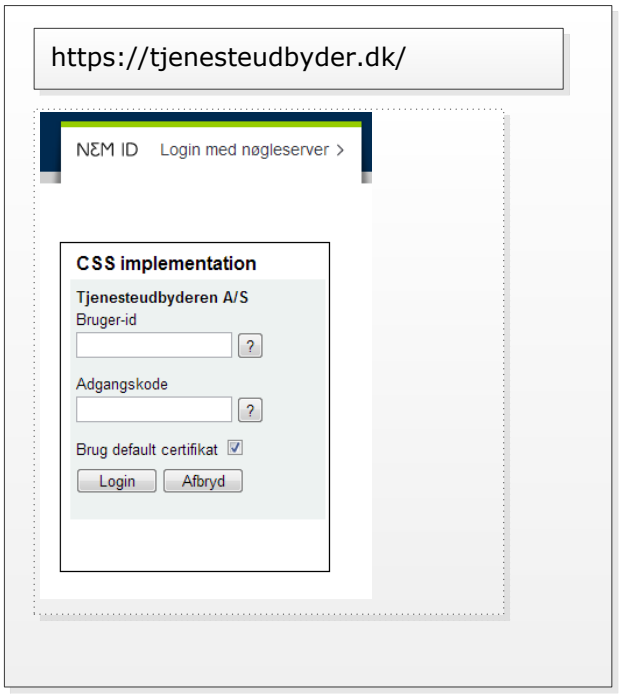

Fig 2: The iFrame

An <iFrame> element does not allow its content to expand beyond its borders, which necessitates that an area sufficient for every possible screen size is allocated, when it is created.

In shorter terms, this means that the iFrame must be created with a fixed width and height.

More technical details on the UX, the flow, parameters etc. are given in the document "Technical specification for LSS".

4 Implementation requirements

In order for the service providers to offer uniform functionality to its end users, regardless of which local signature server the organization of the end user has implemented, common requirements are here laid down for the LSS implementation.

LSS implementations must fulfill these requirements and document this through tests as specified in the document "Test procedures for LSS" and a report to Digitaliseringsstyrelsen, as specified in the document "Requirements feedback form for LSS".

Req. 1	The LSS supplier implementation must support both **authentication and signing**. For signing the formats TEXT, HTML, XML and PDF must be supported.

Req. 2	The implementation must **follow the API** and technical specification given in the document "Technical specification for LSS"

Req. 3	The implementation must be **tested successfully** following the specification given in the document "Test specification for LSS". The results of the tests must be communicated to Digitaliseringsstyrelsen in the template form given in the document "Solution test template for LSS" Further the document "Customer template LSS" must be communicated to Digitaliseringsstyrelsen containing relevant documentation of the implemented functionality minded for the LSS organizations and service providers.

Req. 4	The implementation of the API by the LSS supplier must follow **common standards** for HTML, JavaScript etc. The solution as such must be supported in standard browsers on Apple iOS and OSX, Linux, Android, Microsoft Windows and Windows Phone, without the need for installing extra software etc.

Req. 5	The LSS supplier must describe conditions and prerequisites regarding the **performance** and **scalability** of the solution. Description is communicated in the "Customer template LSS".

Req. 6	All requirement for organizations implementing the suppliers LSS integration must be stated in the "Customer template LSS". This includes requirements for establishing a local DNS service for the bootstrap process and other technical requirements for the solution to work.

Req. 7	The LSS supplier must support the service providers and end users with the best possible error handling. This includes using common error codes and signaling under best practice as defined in the document "Technical specification for LSS". Error messages aimed at end users must be meaningful to the user and suggest context dependent error remedy.

Req. 8	The LSS supplier must document in the "Customer template LSS" how best security of the implemented solution is achieved by the organization hosting the LSS. This includes technical requirements as well as awareness and training requirements. Protection of mobile devices, operations servers and operations environment must be described and required by the LSS supplier.

Req. 9	The relevant requirements from the OCES certificate policy must be satisfied by the LSS implementation.

Req. 10	The LSS implementation must be optimized for performance. Notably the API must be efficient with respect to - fast download - fast startup on device - efficient NemID operations with least network traffic - best possible performance experience for the user, avoiding blocking of GUI updates etc.

Req. 11	The LSS implementation must use standardized NemID dialogues and UX following best practices and recommendations put forward in the LSS for NemID materials. The LSS supplier must document NemID UX in the "Solution test template for LSS" reporting.

Req. 12	The UX implemented by the LSS supplier should follow the usability recommendations put forward in the LSS for NemID documentation, otherwise the LSS supplier must test and document that the usability of the implemented UX is at least on par with the recommendations put forward in the LSS for NemID documentation. Where no recommendations exist, the supplier must ensure usability through own tests.

Req. 13	The LSS implementation must support best possible handicap accessibility.

Req. 14	Requirements from "Persondatalovens foreskrifter" on the protections of personal information and "Sikkerhedsbekendtgørelsen" must be satisfied by the LSS implementation

Req. 15	The LSS supplier must address functional or security related errors in a swift way, and provide updates and assistance to its customers in order to keep the LSS installations as up to date as possible.

Req. 16	The LSS supplier must display the REQUESTISSUER parameter to the end user if CLIENTFLOW is logon.

5 Implementation recommendations

The following section describes recommendations to the implementation of the LSS.

Rec. 1	It is recommended that the UX implemented by the LSS supplier is recognizable by the users as being another NemID / "Digital Signatur" solution. A layout not deviating unnecessarily from the known layout of the existing NemID solutions will help user recognize the solution and workflow. A visual proposal can be found in the section "LSS visual guideline".

Rec. 2	It is recommended that the users are able to find a reference to their local helpdesk inside the iFrame. This information should be available in a similar manner across different LSS supplier's solutions. [Note: to be established] A visual proposal can be found in the document "LSS visual guideline".

Rec. 3	For signing flows, it is highly recommended, that the LSS-supplier utilize the entire allocated space inside the iFrame in a dynamic way, such that the service provider is able to scale the frame according to the screen-size of the user's device.

Rec. 4	The LSS supplier should display the content of the REQUESTISSUER parameter if CLIENTFLOW is signing.

Rec. 5	The LSS supplier should handle as many error situations as possible before returning an error to the service provider.

Rec. 6	The LSS supplier should verify the parameter signature and audit log the subject serial number of the certificate passed in the SP_CERT parameter along with the timestamp.

Rec. 7	The LSS supplier could log the performed signatures

6 LSS visual guideline

This section specifies the visual guidelines given for the LSS implementation.

<To be completed>

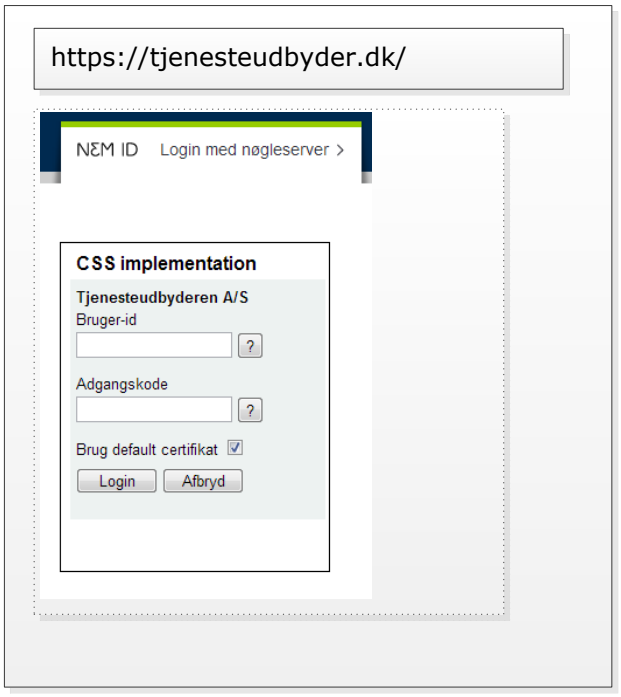

Fig 3: Logon UX with neutral "LSS implementation"

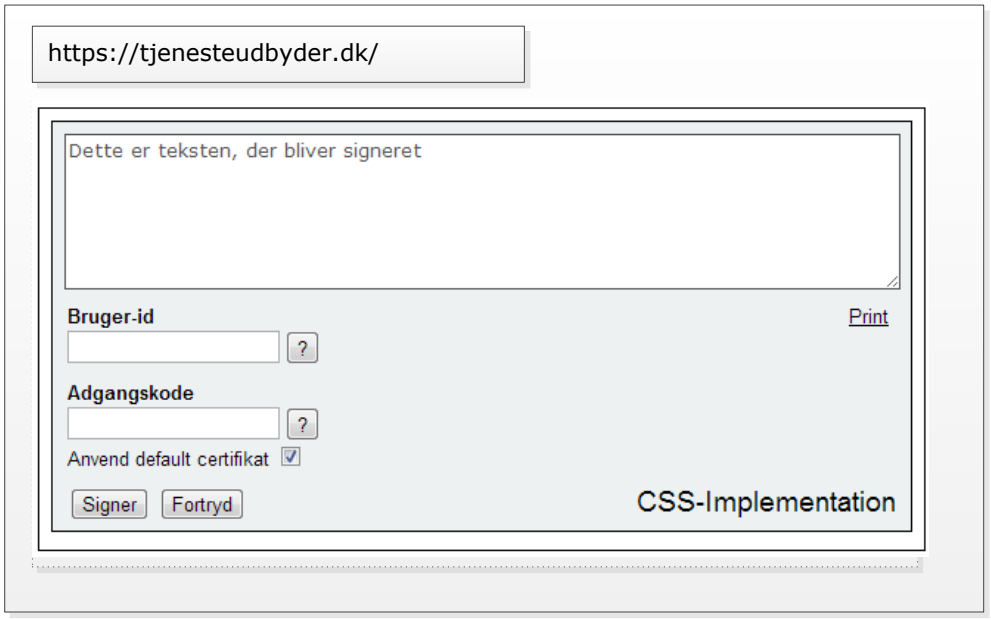

Fig4: Signing UX with neutral "LSS implementation"