Server based signature service

Overview

Based on federated identity

Swedish e-Identification infrastructure

Table of contents

1 INTRODUCTION

This overview describes the basic architecture of the server based signing service that is being deployed in the Swedish national e-ID infrastructure.

Sweden has over the past decade successfully developed public electronic services that are enabled through a national electronic identification infrastructure (e-ID). Two important functions of this infrastructure are to allow citizens to authenticate themselves securely and to electronically sign documents that represent transactions, agreements, applications and declarations of various forms and types.

The Swedish e-Identification board is currently working to upgrade the infrastructure for electronic identification in Sweden, based on federated techniques using the SAML standard. One of the many advantages of such an infrastructure is that users authenticate to the e-services with SAML assertions, which allows service providers a standardized technique to authenticate users regardless of the technology used in the user's electronic ID. This allows a wide variety of e-ID technologies (eg, mobile solutions, smart cards, OTP-generators, etc.) as long as they meet requirements of the level of assurance requirements in the trust framework.

As a consequence, the users e-ID token in the upgraded infrastructure no longer need to contain a private key and certificate in accordance with X.509 standard, which means that such users no longer have the capability to sign using a key and certificate stored on the users e-ID token or local device.

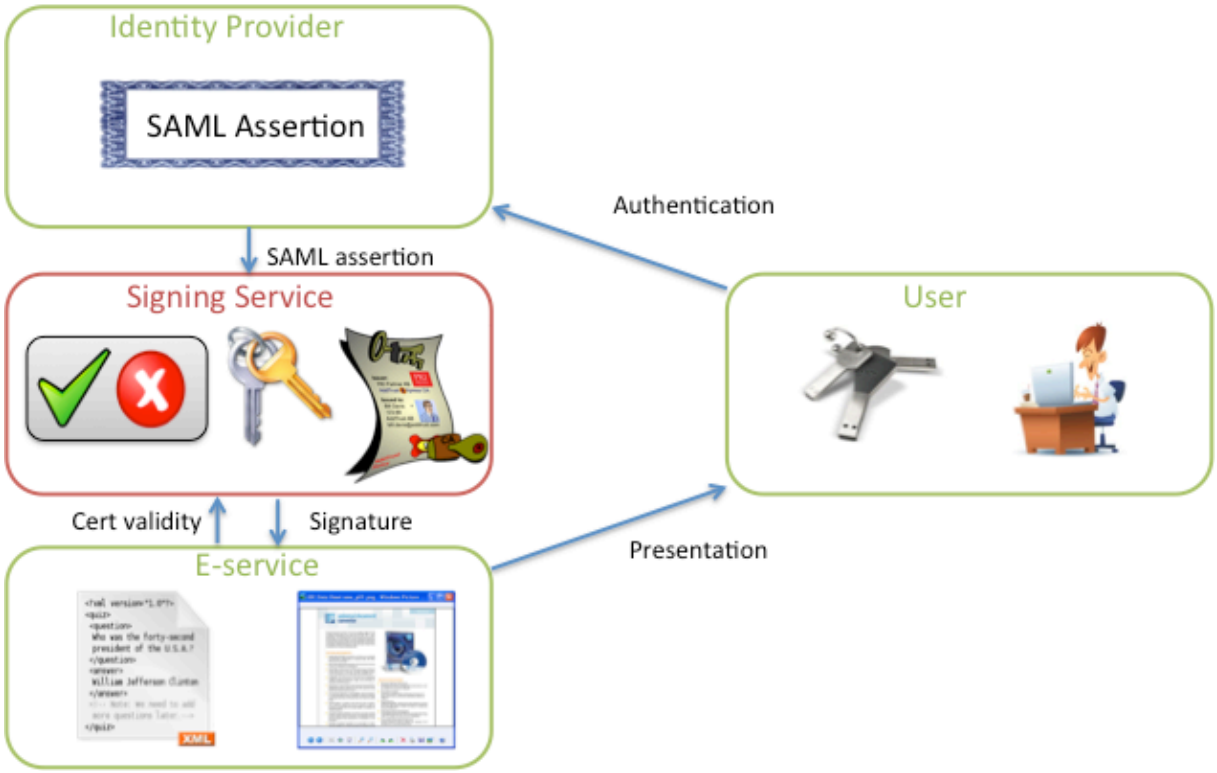
The new server-based signing services are intended to enable citizens to sign electronically in the new infrastructure using any supported and approved authentication.

These signing services enable service providers to allow users in their services to sign documents electronically using the new e-ID infrastructure. This is done by transferring the user together with a standardized sign request to a signing service using an extended profile of the OASIS DSS protocol. The user then authenticates to the signing service in order to sign the document. The signing service does not keep track of users and user keys, but simply generates a new signing key and signature certificate for each instance of signing. This is a procedure that provides a number of important advantages such as:

- It reduces the amount of user sensitive information that needs to be stored on the server and hence, limits the amount of user sensitive information that can be stolen and abused.
- It provides proof of signing time as the certificate is generated at signing time.
- The signature is always associated with a certificate that has sufficient remaining validity time for the intended use of the signed document.
- The certificate content can be adapted to the intended use of the signature such as whether the certificate should contain a private or a professional identity.

2 FUNCTIONAL OVERVIEW

The following picture illustrates the basic functions of a signing service.



The user is in this case logged in to an agency's e-service and has reached the point where the user needs to sign an electronic document, such as a tax declaration in the Tax authority e-service.

The user then signs the electronic document by the following procedure:

- The e-service presents the information that will be signed to the user. This is typically done by viewing function in the e-service with a clear function (button) that the user selects upon agreeing to sign the present information.
- The e-service creates a signed sign request and forwards the user to the signature service of this request.
- The signing service verifies the sign request.
- The signing service authenticate the user that has been requested to sign through the same identity provider that the user used to log into the service provider that sent the sign request. The signing service verifies through the obtained SAML assertion that the authenticated user is the intended signer.
- The signing service generates a signing key and signing certificate and then generates the user's requested electronic signature.

- The signing service creates a sign response using an extended profile of the OASIS DSS protocol, which contains signature and status information, and returns the user to the e-service with this sign response.
- Upon receiving the sign response, the e-service compiles the signed document and verifies the signature.
- The e-service confirms to the user that the document is successfully signed.
- Future verification of the signing certificate is supported by revocation information that is freely available from the signing service.

This model allows the full document to be signed to be included in the sign request, but this is not how this is implemented in the Swedish eID infrastructure. Instead, just the canonical signed info (XML) or signed attributes (CMS/PDF) are sent to the signing service for signature generation. This allows the requesting service to have a signature created without revealing the information to be signed to the central signing service. This is an important requirement for many government agencies that are required by law to protect sensitive information about citizens.

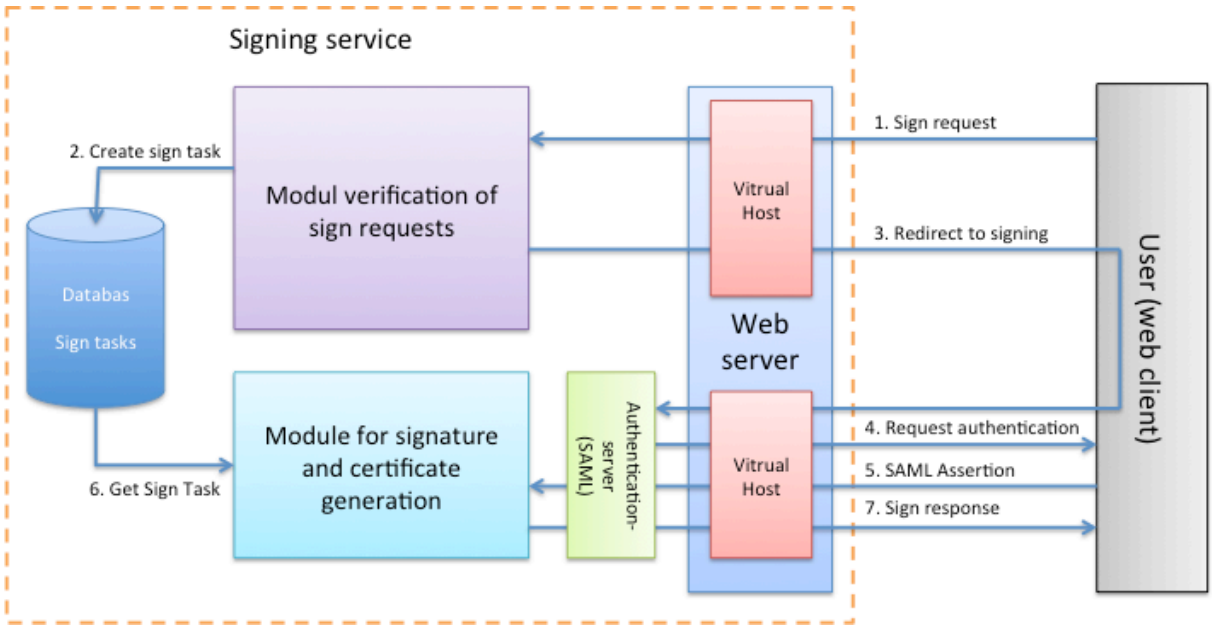
In the implementation of these signing services with government e-services, the E-government service is responsible towards the user for the provided signing service. This is also reflected in the federation metadata so the a user that authenticates to the signing service using its own selected e-ID and associated identity provider, with see that they authenticate to a signing service that belongs to the same authority that requested the signature.

In practice the signing services are provided by contracted approved third party trust services that provides an instance of a signing service to each government authority, unless for some reason a government authority decides to setup their own local signing service.

Many individual instances of a signing service usually use a common certificate issuing service (CA) chained to a common trusted root in order to support global trust in generated signatures, potentially supported by Trust service Status Lists (TSL).

After successfully generating a signature the signing service can safely delete all records of the generated signature with exception of the assigned certificate serial number that is kept to allow certificate revocation. All necessary evidence of the signing is placed in the signed sign response or in the generated signature or signature certificate.

The internal process of the signing service is illustrated in the following figure:



This illustrates the steps in the signature process as seen from the signing service's perspective:

1. The signing service receives the sign request from the user's browser. Transfer of the user with the attached request for signature is using a technique that is equivalent to the HTTP POST binding of the SAML protocol specification.
2. The sign request is checked. If the request for signature is OK, a signature task is created and stored in an internal database.
3. The user is transferred (redirected) to a protected URL of the signing service, which enforce that the signer is authenticated using a valid SAML assertion before being allowed access to the protected resource for signature generation.
4. The authentication server that is adapted to enforce SAML authentication redirects the user the identity provider identified in the request URL. The assigned identity provider is obtained from the verified sign request.
5. The identity provider returns the user to the authentication server with a SAML assertion. The user is allowed access to the protected resources for signature creation upon successful verification of the assertion.
6. The signing service verifies the authenticated user is the intended signing and generates a key, certificate and the requested signature.
7. The signing service creates and signs a sign response and transfers the user back to the requesting service provider with the sign response.

Optionally, this process may have an intermediary step between step 2 and 3, where the user is prompted by the signing service that the user is about to sign where the signing service may present information about the signature this is going to be created. This may include a sign message encoded in the sign request that was obtained from the service provider.

3 SIGN SUPPORT SERVICE

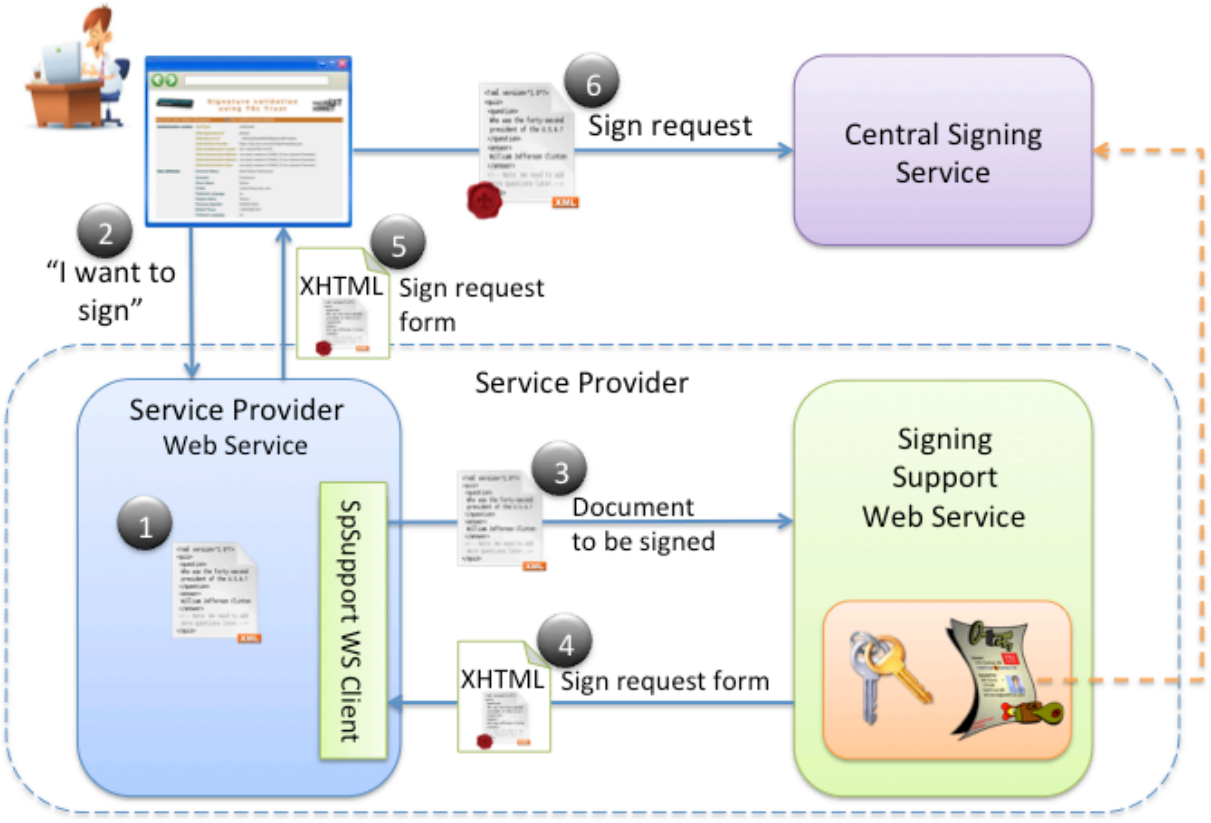
The service provider will have to generate sign requests and to process sign responses in order to construct signed documents. This involves advanced signature operations that are challenging to implement correctly for a service provider.

This infrastructure therefore also includes a standardized web service API to a local sign support service that is deployed locally at the service provider local network.

This web service API has 3 operations:

- Sign-request – for generating sing responses
- Complete-signing – for receiving sign responses and compiling signed documents
- Signature verification – for verifying a signed document.

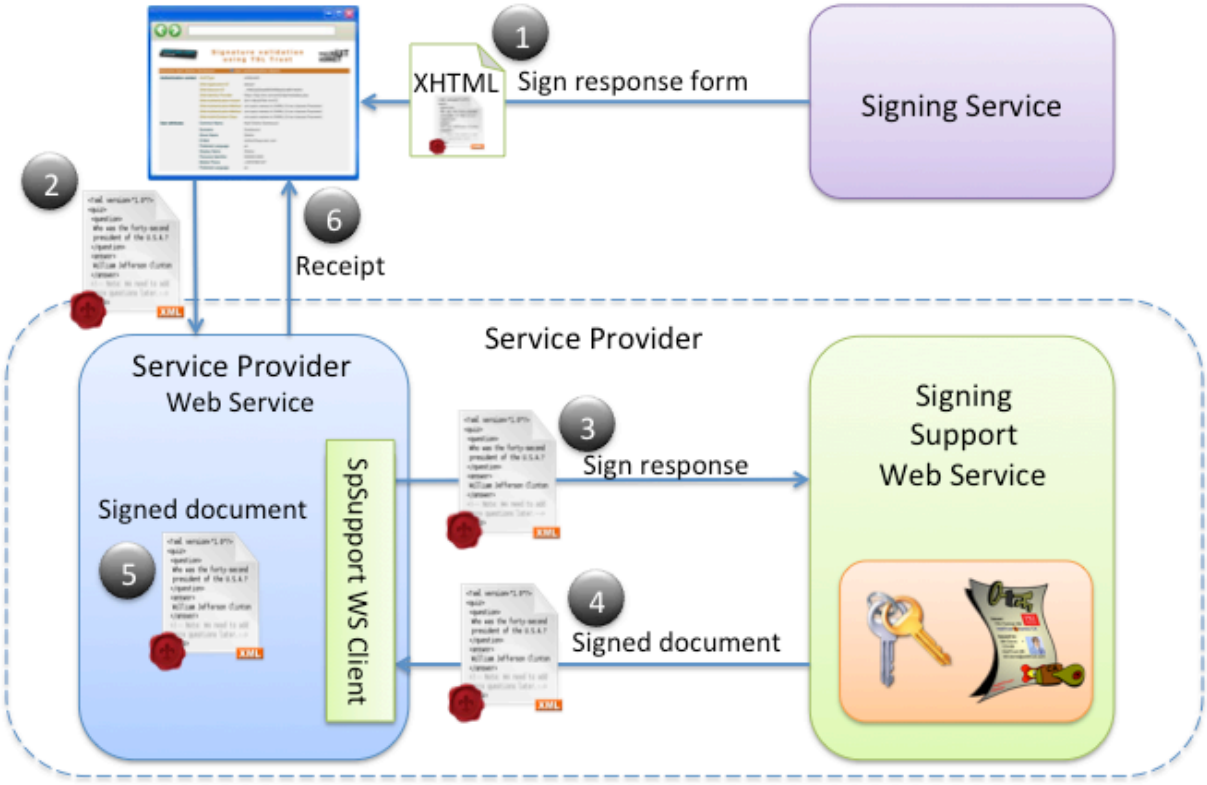
The pre-signing phase is illustrated in the following figure:



1. The service provider has an electronic document that the user needs to sign. The document is presented to the user.
2. The user accepts to sign
3. The document to be signed is sent to the signing support service using a sign-request operation in the WS-API.
4. An XHTML page is returned with an embedded sign request.
5. The XHTML page is returned to the user.

6. As the users web client renders the XHTML page, the sign request is posted to the signing service.

The complete-signing process is illustrated by the following figure:

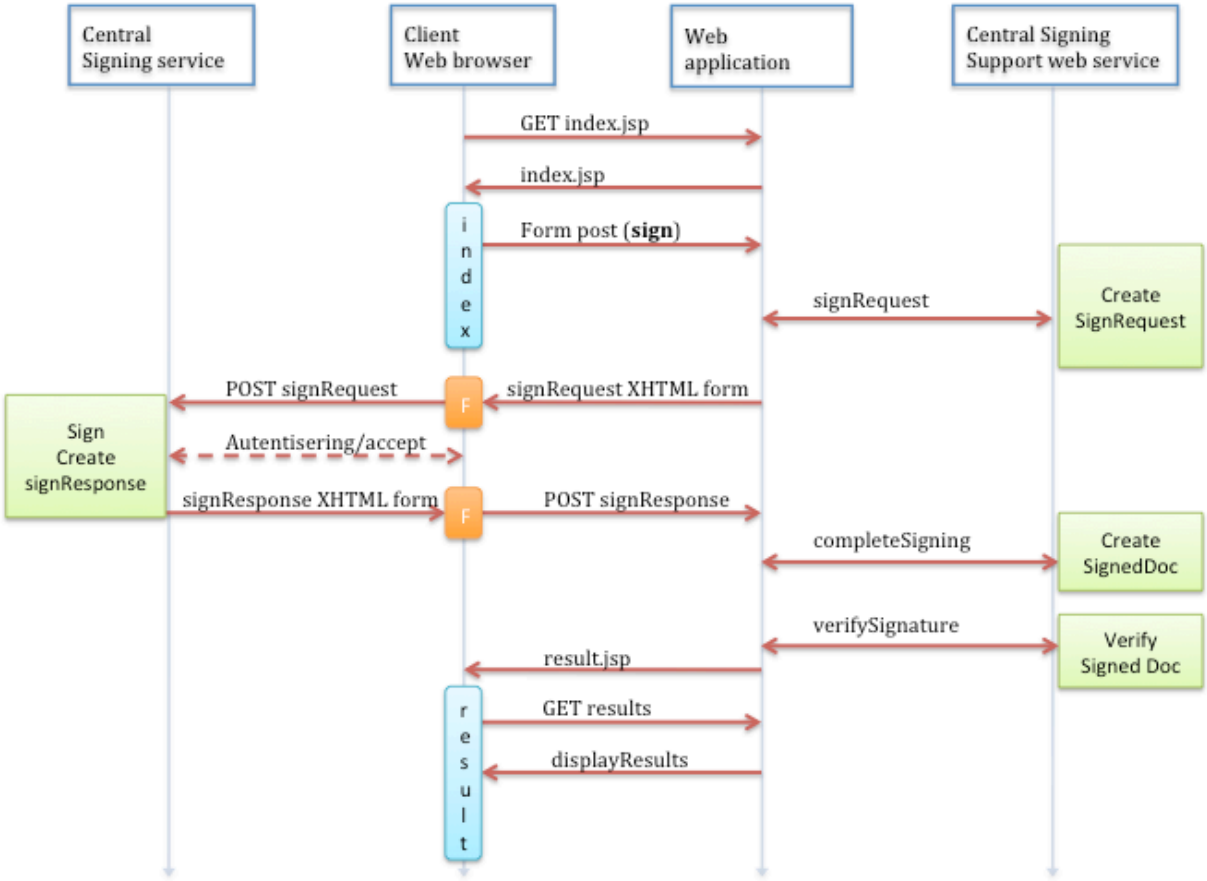


1. After successful completed signature generation, the sign service returns an XHTML page to the user with an embedded sign response.
2. As the users web client renders this XHTML page, the sign response is posted to the service provider.
3. The sign response is sent to the signing support web service using the WS-API operation for complete signing.
4. The support service compiles the signed document and returns this to the service provider as a response message to the complete signing API call.
5. The service provider validates the signed document (This may be done using the signature verify operation in the WS-API)
6. The service provider acknowledges to the user that the document has been successfully signed.

The WS-API is documented in a WSDL file that is made available form the web service. This WSDL file is used to build the WS Client at the service provider end for calling the operations provided by the web service.

4 SEQUENCE DIAGRAM

The following diagram illustrates an example of a complete flow of messages in a signing process:



- The user accesses a service by obtaining a web page providing the service.
- This service web page contains functions for signing some data. As the user accepts to sign an http request is sent to the service provider.
- The service provider use the WS-API to obtain a sign request from the support service.
- The service provider returns the XHTML page with the sign request to the user web client. A java script in the XHTML page causes the sign request to be posted to the sign service.
- The sign service verifies the request, authenticates the user and generates the signature.
- The sign service returns a sign response embedded in a XHTML page. A java script in the page causes the web client to post the sign response back to the service provider.
- The service provider obtains the signed document from the support service by providing the sign response in a complete signing operation in the WS-API.
- The same WS-API is used to verify the signed document.

- The service provider returns a result page where the user can select and view various result and status information.

5 RESOURCES

Protocol specifications, XML schemas and link to source code are currently available through: http://aaa-sec.com/eid2/sigsupport