TU Services flow and security

Version 1.1.0

Signaturgruppen A/S
INCUBA Navitas, Inge Lehmanns Gade 10, 8000 Aarhus C
www.signaturgruppen.dk
info@signaturgruppen.dk

Signaturgruppens TU Services flow and security

Introduction

This document is intended as a reference for the flow and protocols for Signaturgruppen TU Services and for the security of the communication and data affected.

Overview

The solution consists of two parts:

1. A client library, which has to be integrated into your website, in the following denoted as the TU Services Client (SPS.Client).
2. A web application working as backend for your website(s) integrating with NemID, in the following denoted as the TU Services Backend (SPS.Backend).

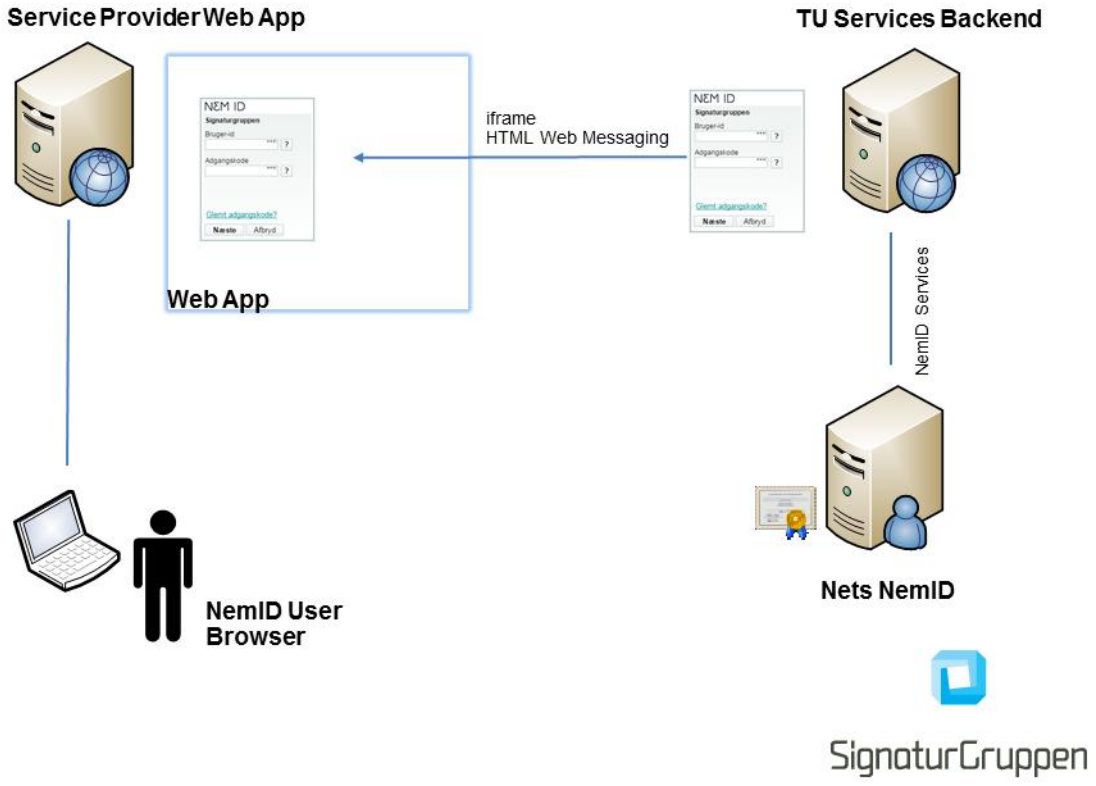

Figure 1. Overview of TU Services setup

Flow diagram

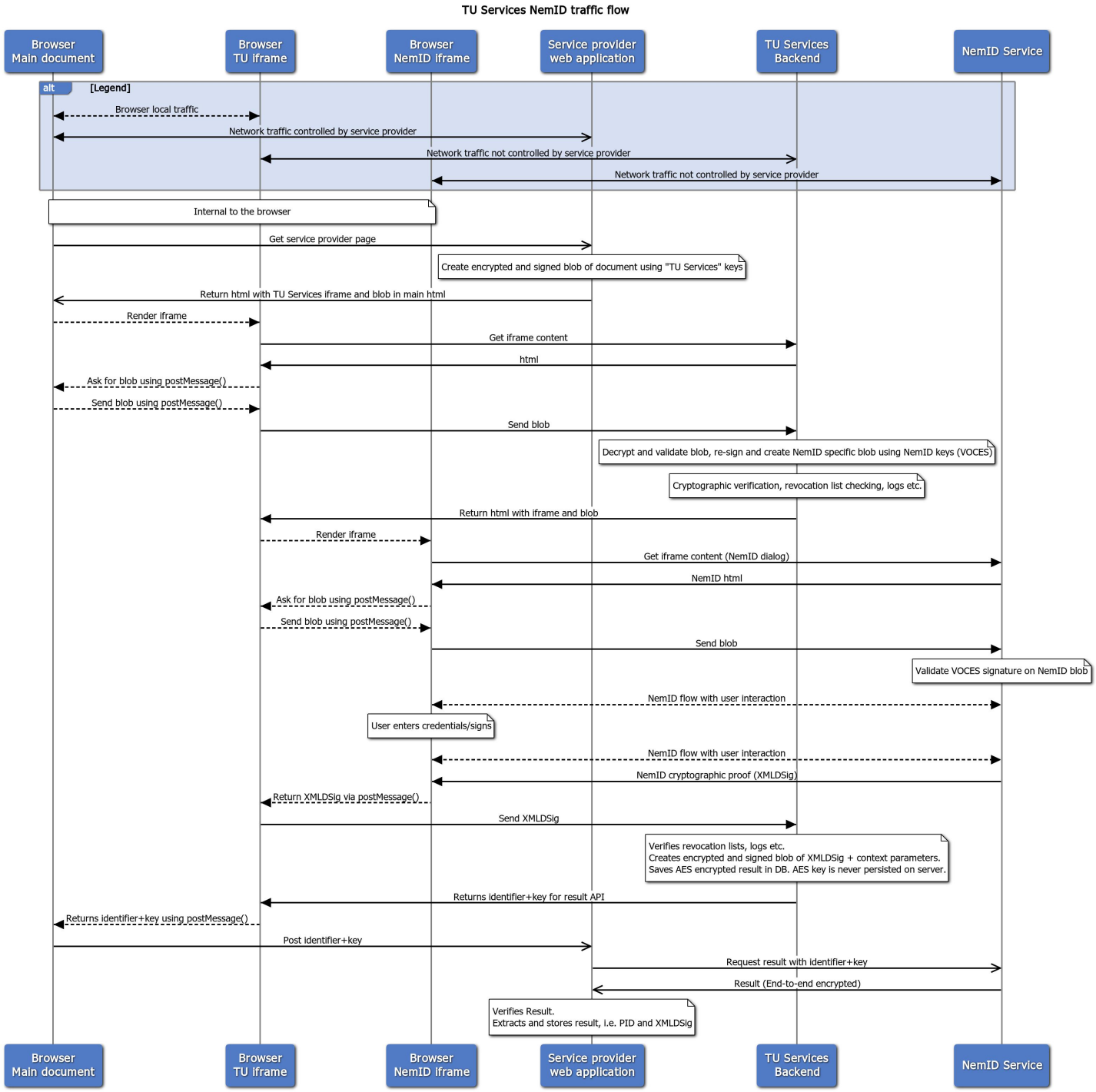

Figure 2. Flow diagram of the TU Services iframe flow

The following shows the flow of the TU Services iframe based protocols

All communication between the TU Services Client and the TU Services Backend is based on JavaScript sent through an iframe through a secure channel (HTTPS) and furthermore signed and encrypted using RSA-2048/SHA-256/AES-256.

The client is used to sign and encrypt a JSON parameter payload (2), which is sent through the iframe to the TU Services Backend (3) using the Web Messaging JavaScript API or using the service API available.

The backend verifies the payload and decrypts the data (4) then setup the requested flow for the client (5).

When the flow completes (6+7), the backend verifies the result from the flow (8) and then saves the encrypted result in the database and sends the identifier + encryption key for the result to the TU Services Client (9+10+11).

The client uses the identifier + key to retrieve the result from the backend over an end-to-end secured channel (12+13).

The client verifies and decrypts the payload and creates a structured result, which gives the control back to the TU web application (14).

Pinned communication

The client and backend communicates using a set of pinned RSA-2048 public/private keys (each have their own private key), which ensures that the communication is secured at application level end-to-end in addition to communicating through HTTPS.

Sensitive data

For all flows included in the standard TU Services setup, which includes login and signatures with NemID and all associated services, the backend only stores the required logs to the TU services backend database (see later section).

For signing flows, this means that the document signed by the user is accessible to the backend but only used for verification of the response at the end of the flow and is not persisted to disk or database by the TU services backend.

Persistent data about the service provider (TU)

All sensitive TU information, including the company certificate (VOCES) and corresponding private key used by the backend to sign the parameters for the NemID flow, are encrypted in the database and only accessible at runtime by the backend application with the correct decryption key embedded.

Communicated data

The iframe flow is based on signed and encrypted JSON parameter blocks: One JSON block for the setup of the flow initiated from the TU Services Client and one JSON block for the response initiated from the TU Services Backend.

For login flows the JSON parameter block sent from the TU Services Client to the TU Services Backend contains only the requested flow parameters. These includes: type of flow, type of client (responsive/normal), timestamp, client ID, and a signature of the parameter block. No user-related information is sent in the request.

For signing flows, the document-to-be-signed is included as one of the parameters in encrypted form.

The TU Servies Backend verifies the signed parameter block, decrypts the encrypted data and starts the requested operation (if allowed).

When the NemID flow completes, the TU Services Backend receives either an errorcode or the signed XMLDSig (the NemID signature XML - same for both login and signing scenarios).

The TU Services Backend verifies the XMLDSig, logs required entries (see below) and creates a signed and encrypted JSON response to the TU Services Client.

Example of login request parameters:

```
{"BodyTypeName":"Frame.NemID.NemIDFlowConfig",
"BodySerialized":
"{\"UseLimitedMode\":true,
\"RememberUserIdToken\":null,
\"Language\":\"Da\",
\"SignProperties\":[],
\"ClientFlow\":\"NemID\",
\"Timestamp\":null,
\"TransactionIdentifier\":null}",
"ClientIdentifier":"7ec45e5a-e0ad-48c7-a2d5-1f41350191fb",
"TransactionIdentifier":"ecf8f940-ab4d-4829-8619-b610f9b5130a",
"Timestamp":"1478872876207",
"EncryptedKey":null,
"Signature":"...",
"ClientInfo":{"xxx"}}
```


For signing flows, the JSON response contains the same as for login + the document to be signed in encrypted format.

Example of response parameters:


```
{"BodyTypeName":"Frame.NemID.NemIDFlowResult",
"BodySerialized":"...",
"ClientIdentifier":"7ec45e5a-e0ad-48c7-a2d5-1f41350191fb",
"TransactionIdentifier":"ecf8f940-ab4d-4829-8619-b610f9b5130a",
"Timestamp":"1478873125041",
"EncryptedKey":"...",
"Signature":"...",
"ClientInfo":{"xxxx"}}
```


Here BodySerialized contains the encrypted response, which for NemID flows is the XMLDSig. The EncryptedKey is the encrypted AES key used for encrypting the message.

PID/CPR API Service

For the PID-CPR-MATCH API service available in the TU Services Client, the request and response parameters are of the same form as described above.

The CPR is encrypted and used only to call the NETS service. The CPR is never persisted and not included in the response to the TU Services Client.

Logged entries

For all NemID flows we log the following: certificate info including PID/RID, public certificate info, revocation check info, certificate issuer chain and the SHA-256 digest of the XMLDSig.

We do not log any of the signed attributes of the XMLDSig document nor the signed document.

For the NemID PID-CPR-MATCH API service call we log the PID and TU Service Client together with the result of the call (true/false). We do not log the CPR.

Confidentiality of NemID

All communication in the NemID variants flows through the users browser which facilitates the transfer of data between the service provider website and the backend through the iframe using the Web Messaging API or through an end-to-end secured API directly between the TU Services Client and the TU Services Backend.

The payloads flowing between the client and the backend are signed and/or encrypted (sensitive information), and thus no sensitive data is available to the user or any plugins/applications running on the users device.

For NemID flows the NemID client is started inside the SPS iframe which is itself secured and running HTTPS. Neither the backend or the service provider website is able to read or see the communication between the user and the NemID backend servers.

The security for the NemID client is the same in this setup as for standard NemID integrations as the NemID client creates a secure end-to-end channel between the user and the NemID backend servers.

Contact information

Please do not hesitate to contact us if you have any questions.

Info	
Mail	support@signaturgruppen.dk
Phone	+45 70256425
Website	http://www.signaturgruppen.dk