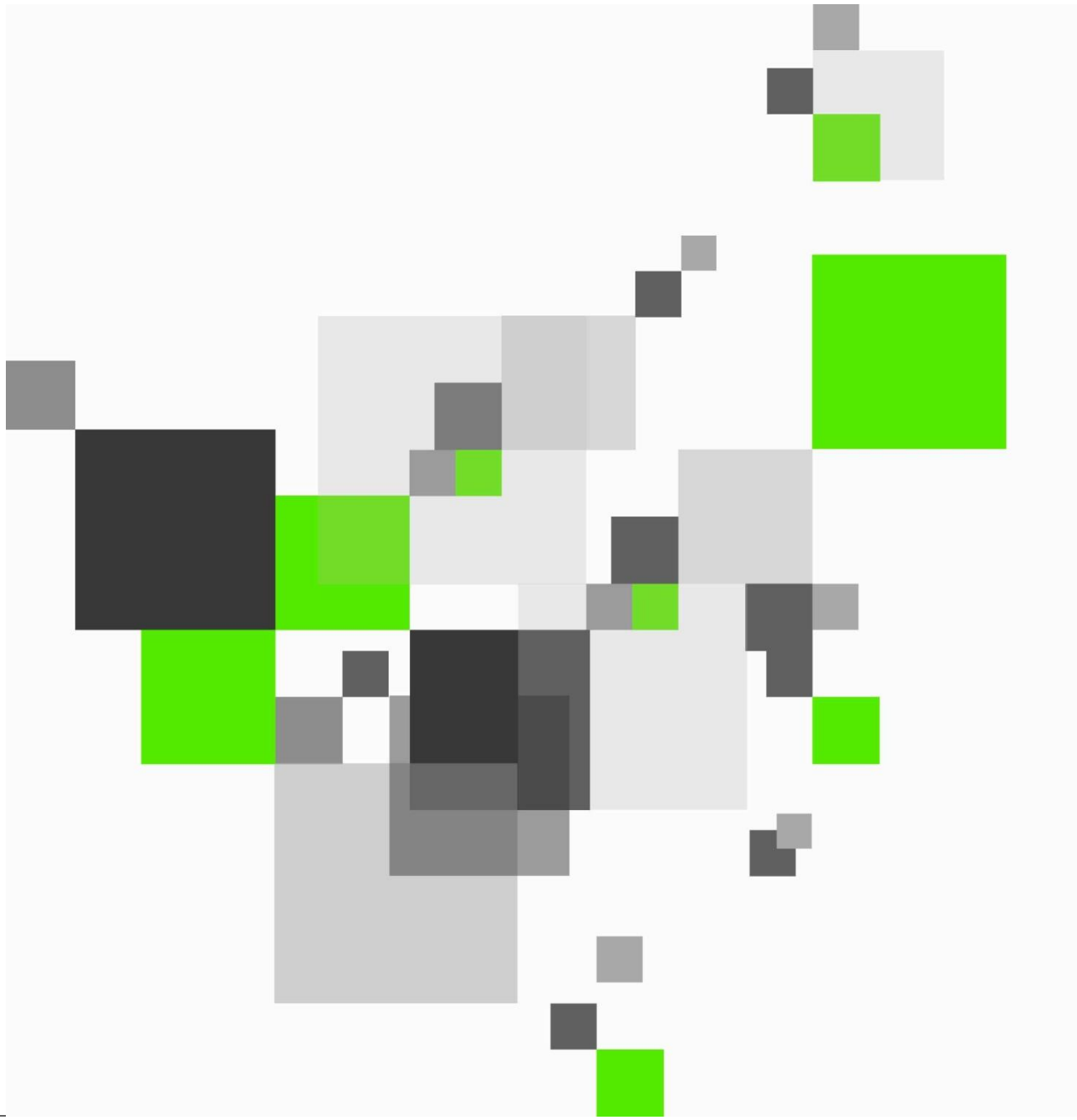
Security Token Service –

Snitflade OnBehalfOf Proxy



Indholdsfortegnelse

1. Versionsnummer

Snitfladens version er 1.0.

2. Snitfladebeskrivelse

Dokumentet beskriver en snitflade hvor et Anvendersystem kan forespørge en SAML2.0 assertion på vegne af et andet Anvendersystem.

Snitfladen understøttes af Security Token Service (STS) støttesystemet.

Snitfladen er baseret på OIO WS-Trust Profile v1 0 1 [WST-OIO], og følger i høj grad denne. Snitfladen indeholder dog også elementer der afviger væsentligt. På områder hvor dette dokument er mere specifikt end [WST-OIO], er dette dokument gældende. Hvis der er uoverensstemmelse imellem dette dokument og [WST-OIO], og dette dokument ikke er overholder [WST-OIO], så er [WST-OIO] gældendende.

Snitfladen understøtter den WS-Trust baserede autentifikations forespørgselsprotokol, hvor en service provider sender et token request til Security Token Service i form af et RequestSecurityToken element (RST), som så behandles af Security Token Service, og et RequestSecurityTokenResponse bliver sendt indeholdende et signeret SAML2.0 assertion tilbage til klienten.

Snitfladen er forskellig fra [SnitfladeBeskrivelse-STS-OIO WS Trust] da kalderen forespørger et token på vegne af et andet Anvendersystem, og fordi der udstedet et token med et OnBehalfOf element.

1.1 Use Case

Denne snitflade understøtter følgende use case:

1.1.1 Definitioner

Eksternt system = ES
Serviceplatformen = SP
Fælleskommunal service = FS

1.1.2 Trin

1. ES ønsker at forespørge FS, men er ikke en del af den fælleskommunale platform. ES forespørger FS, ved at kalde en snitflade for FS der udstilles vha. SP.
2. SP forespørger et token hos STS på vegne af ES. SP autentificerer sig over for STS vha. SP eget OCES certifikat, og indlejrer den offentlige del af ES' certifikat for at fortælle STS hvilket Anvendersystem der forespørges et token for.
    a. ES er oprettet som Anvendersystem i STS ved at være provisioneret fra Støttesystem Administrationsmodul.
3. Når SP har fået udstedt et token på vegne af ES, så kalder SP FS og sender svaret tilbage til ES.

3. Servicebeskrivelse

3.1 Identity provider

Anvendersystemet og Security Token Service SKAL have udvekslet metadata før Security Token Service vil acceptere et RequestSecurityToken fra **anvendersystemet**.

Metadata for servicen er adressen og vil blive uploaded til støttesystem Administrationsmodul og derfra provisioneret til Security Token Service.

Anvendersystemet er identificeret ved et eller flere OCES certifikater. Anvendersystemer der skal kunne forespørges tokens på vegne af, er ligeledes oprettet som Anvendersystemer.

3.2 Supported binding

Liberty Basic SOAP Binding.
HTTP over TLS SKAL anvendes. Der anvendes IKKE klientcertifikat med TLS.

3.3 Signering af request

Elementet <wst:RequestSecurityToken> SKAL signeres af den der laver forespørgslen.

Der SKAL benyttes det VOCES eller FOCES certifikat der på forhånd er registreret i Administrationsportalen, til at foretage signeringen.

3.4 Krav til <RequestSecurityToken>

Elementet `<wst:RequestType>` SKAL indikere udstedelses binding og derfor skal følgende URI anvendes: `http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue`.

Elementet `<wsp:AppliesTo>` SKAL indeholde en `<wsa:EndpointReference>` der identificere den web service udbyder eller den specifikke service som identitetstokenet skal udstedes for.

3.5 Inputparametre (RST request element)

Det er nødvendigt for kalderen at ”fortælle” STS hvilken Anvenderkontekst som kalderen benytter. Måden dette er implementeret på, er ved at kalderen, i listen med requested claims, tilføjer et claim med en specifik claim type, der angiver den Anvenderkontekst der benyttes.

Derudover er det nødvendigt for kalderen at ”fortælle” hvilket Anvendersystem der forespørges et token på vegne af. Normalt gøres dette, ved at indlejre et token for Anvendersystemet der forespørges på vegne af, men da der ikke findes et sådan med denne binding, så indlejres i stedet et certifikat der repræsenterer Anvendersystemet [der forespørges på vegne af].

Dette certifikat indlejres ved at tilføje et claim til listen med forespurgte claims, der angiver certifikatet.

Parameter	Obligatorisk	Udfaldsrum
HTTP METHOD	*	POST
RequestSecurityToken	*	Forespørgsel for at modtage et token
RequestType	*	Indikation af ”class of function” som bliver forespurgt. Her skal benyttes http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue jf. afsnit 3.4
AppliesTo	*	Specificerer scope som token skal overholde. Jf afsnit 3.4 skal <wsa:EndpointReference> benyttes
TokenType	*	Beskriver typen af sikkerhedstoken’et, specificeret som en URI. Skal sættes til: `http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0`
OnBehalfOf		Value skal være et base64 encodet OCES certifikat der repræsenterer det Anvendersystem der forespørges token for. Certifikatet skal på forhånd være registreret via Administrationsmodulet, og kalderen skal på forhånd være givet rettigheder til at forespørge OnBehalfOf.
Claims – Anvenderkontekst	*	Præcist eet ClaimType element i Claims listen, der repræsenterer den Anvenderkontekst der arbejdes i. Claim type skal være: ”dk:gov:saml:attribute:CvrNumberIdentifier” og claim value skal være CVR for den benyttede Anvenderkontekst

Eksempel på et RST element:

```
<S11:Envelope xmlns:S11="..." xmlns:wsu="..." xmlns:wsse="..." xmlns:xenc="..."
xmlns:wst="...">
     <S11:Header>
          <wsa:Action>http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</wsa:Action>
          <wsa:MessageID>urn:uuid:99999999-0000-0000….</wsa:MessageID>
          <wsa:To>http://sts.example.org</wsa:To>
          <wsse:Security mustUnderstand=”1”>
               <wsu:Timestamp wsu:ID=”ts”> … </wsu:Timestamp>
               <ds:Signature xmlns:ds="...">
                  <ds:SignedInfo> …
                     <ds:Reference URI="#req"> … </ds:Reference>
                     <ds:Reference URI=”#ts”> ... </ds:Reference>
                     <!-- More references to other header elements -->
                  </ds:SignedInfo>
                  <ds:SignatureValue> … </ds:SignatureValue>
                  <ds:KeyInfo>
```

```
                    <ds:X509Data> … sender certificate … </ds:X509Data>
                  </ds:KeyInfo>
                </ds:Signature>
          </wsse:Security>
     </S11:Header>
     <S11:Body wsu:Id="req">
          <wst:RequestSecurityToken Context=”urn:uuid:00000…”>
               <wst:TokenType>
                    http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0
               </wst:TokenType>
               <wst:RequestType>
                    http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
               </wst:RequestType>
               <wst14:ActAs>
                    <!-- Include token for user that is acted on behalf of here -->
               </wst14:ActAs>
               <wsp:AppliesTo>
                    <wsa:EndpointReference>
                         <wsa:Address>http://agency_x.dk</wsa:Address>
                    </wsa:EndpointReference>
               </wsp:AppliesTo>
               <wst:Claims wst:Dialect=”http://docs.oasis-
               open.org/wsfed/authorization/200706/authclaims”
               xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
                  <auth:ClaimType Uri="dk:gov:saml:attribute:CvrNumberIdentifier" Option-
               al="false">
                    <auth:Value>12345678</auth:Value>
                  </auth:ClaimType>
               </wst:Claims>
               <wst:OnBehalfOf>
                    [...base64 encoded X.509v3 certificate...]
               </wst:OnBehalfOf>
          </wst:RequestSecurityToken>
     </S11:Body>
</S11:Envelope>
```

3.6 Output – RSTR Response

Efter brugerautentifikationen, sendes som svar på det modtagne RST element, et RSTR svar som indeholder præcis eet <RequestSecurityTokenResponseCollection> element med præcis eet <RequestSecurityTokenResponse> element.

Parameter	Obligatorisk	Udfaldsrum
HTTP METHOD	*	POST
RequestSecurityToken-Response	*	Indeholder et RequestedSecurityToken indeholder med SAML2.0 assertion. KOMBIT Attribut Profil v1.0 benyttes. Den udstedte assertion indeholder altid en attribut med brugerens anvenderkontekst. Denne anvenderkontekst modsvarer den anvenderkontekst som kalderen sendte med i RST.
AppliesTo		Indeholder <wsa:EndpointReference> fra forespørgslen

3.6 Binding

RSTR returneres ved brug af HTTP POST binding.
Kun HTTP på TLS (HTTPS) tillades.

3.7 Signering

Security Token Service signerer svaret og det udstedte"SAML 2.0 assertion" indeholdende attributter om brugeren.

Det udstedte "assertion" SKAL indeholde et SubjectConfirmation element med en "holder-of-key" reference til forespørgerens nøgle.

3.8 Fejlbeskeder

Hvis der opstår en fejl under behandling af request security token beskeden så returneres en SOAP fault besked til kalderen. Denne SOAP fault indeholder en detaljeret besked om fejlen der opstod under behandling af forespørgslen.

Der understøttes følgende fejlkoder:

Fejlkode	**Beskrivelse**
100	Uventet fejl. Fejl hvor årsagen ikke er kendt.
101	Den anvendte konfiguration kendes ikke. Kalderen forespørger et token for en service STS ikke kender, eller som en kalder STS ikke kender.
103	Forkert formateret forespørgsel. Der er en fejl i forespørgslen.
104	Der forespørges et endpoint der ikke findes.
106	Der opstod en transaktionsfejl under commit af auditlog.
110	Ikke understøttet. Der benyttes en endpoint konfiguration som ikke understøttes.
111	Konfigurationsfejl. Der er en fejl i konfigurationen, der gør at forespørgslen ikke kan behandles.
130	Databasefejl. Der er opstået en fejl ved kommunikation/adgang til den underliggende database.

4. Teknisk beskrivelse

Snitfladen er implementeret i Safewhere*Identify produktet.

5. Kommunikation

Snitfladen anvender HTTPS for at sikre meddelelses fortrolighed og integritet.

6. Transformationsmekanismer

N/A

7. Eventuelt yderligere oplysninger

http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.wsdl