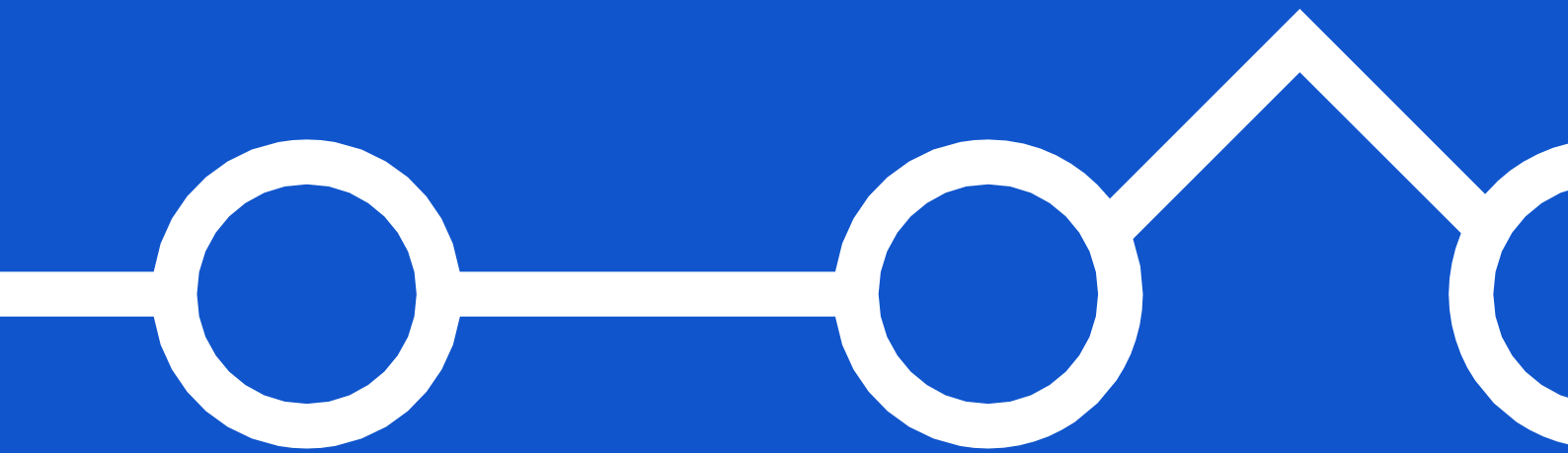


# DIGITALISERINGS KATALOGET

## EGEN IDENTITY PROVIDER

Adgangsstyring for brugere



**KOMB:T**

Kommunernes it-fællesskab

<b>1. INTRODUKTION</b> .....	<b>4</b>
1.1. Forbehold.....	5
1.2. Online Identity Providers.....	5
1.3. Den fælleskommunale sikkerhedsmodel .....	6
1.4. Lokal IdP til intern udvikling og test .....	7
1.5. Tilslutning af egen IdP til føderationen .....	7
1.6. Certifikater.....	8
1.7. Kort om roller .....	8
<b>2. PHP OG NGINX</b> .....	<b>10</b>
2.1. PHP.....	10
2.2. Nginx.....	10
<b>3. SIMPLESAMLPHP</b> .....	<b>11</b>
3.1. Installation.....	11
3.2. Certifikater.....	12
3.3. _include.php .....	12
3.4. Config.php.....	12
3.5. saml20-idp-hosted.php .....	13
3.6. authsources.php .....	14
3.7. KDIAuthAPI.php.....	14
3.8. Opsætning tjek.....	16
<b>4. BRUGERKATALOG</b> .....	<b>19</b>
4.1. Installation.....	19
4.2. Config.php.....	19
4.3. Setup.php.....	19
4.4. Roller og brugere .....	20
<b>5. TEST AF IDP OPSÆTNING</b> .....	<b>20</b>
<b>6. ANVENDELSE I LOKAL FØDERATION</b> .....	<b>22</b>
6.1. Introduktion .....	22
6.2. Registrering af Service Provider i IdP .....	22

6.3. Registrering af IdP i Service Provider.....	23
6.4. Test.....	23
<b>7. TILSLUTTET DEN FÆLLESKOMMUNALE FØDERATION .....</b>	<b>25</b>
7.1. Introduktion .....	25
7.2. Registrering af IdP i føderationen.....	25
7.3. Registrering af føderationsaftale .....	27
7.4. Registrering af Jobfunktionsrolle .....	27
7.5. Registrering af Context Handler i IdP.....	29
7.6. Test.....	30
<b>8. DATAAFGRÆNSNING PÅ ROLLER.....</b>	<b>31</b>
8.1. Introduktion .....	31
8.2. Statiske værdier.....	32
8.3. Dynamiske værdier.....	34
<b>VERSIONSHISTORIK.....</b>	<b>36</b>

## 1. INTRODUKTION

Denne vejledning henvender sig til leverandører, der skal implementere Adgangsstyring for Brugere i den fælleskommunale infrastruktur, dvs. tilslutte deres fagsystem som *Brugervendt system*. Der findes en [vejledning og tilhørende video](#), der forklarer hvorledes du gør dette. Når du har tilsluttet dit brugervendte system og efterfølgende skal teste og benytte adgangsstyringen, da skal du have tilgang til test-brugere i en Identity Provider (IdP), hvilket er emnet for denne vejledning.

Ikke alle kommuner har en fungerende IdP i ExtTest, og det kan være svært, at få etableret kontakt og få prioriteret oprettelse af testbrugere hos kommunen. Det kan også være uhensigtsmæssigt for kommunen, at oprette testbrugere for leverandører, da visse benytter deres produktions-AD som brugerkatalog. Ydermere kan det være, at du ofte i udviklingsfasen har behov for at ændre i din sikkerhedsmodel, efterhånden som dit produkt modnes, så du har ofte brug for at ændre i roller og brugere. Det kan derfor være fordelagtigt for dig, at etablere din egen IdP. Der er to brugsscenerier:

1. Du etablerer en IdP til intern udvikling og test i lokal føderation
  - a. Den fungerer som Context Handler og "emulerer" denne, dvs. konfigureres til at returnere brugers Brugersystemroller.
  - b. Løsningen fungerer kun med egne interne brugervendte systemer.
  - c. Du kan oprette brugere tilknyttet vilkårlige myndigheder, der er ingen begrænsninger.
2. Du etablerer en IdP, som du tilslutter den fælleskommunale føderation
  - a. Du skal være oprettet som Leverandørmyndighed.
  - b. Du tilslutter din IdP og opretter føderationsaftale\* for din egen myndighed.
  - c. Den konfigureres til at returnere brugers Jobfunktionsroller.
  - d. Du kan (bør\*) kun oprette brugere tilknyttet din egen myndighed.
  - e. Den er adgangsgivende til FK KLA, FK ORG og SAPA-P, på vegne af din myndighed.

(\* ) Du bør kun anmode om føderationsaftale til din egen myndighed.

Fælles for dem begge er, at du kan gøre det hele selv, og de kan etableres på en dag, hvis alle forudsætninger er på plads. Så du kan komme hurtigt i gang.

Du kan også vælge begge scenarier. Hvis du vælger at blive oprettet som Leverandørmyndighed og tilmelde din egen IdP til føderationen, da kan det stadig give mening, også at oprette en lokal IdP til intern udvikling og test. Da den lokale variant giver dig mulighed for at teste på vegne af alle myndigheder.

Bemærk: Du kan benytte den samme IdP til begge scenarier. Du gør blot følgende:

- Oprettet et sæt brugere til anvendelse i din lokale føderation, der tilknyttes Brugersystemroller
- Oprettet et sæt brugere til anvendelse i den fælleskommunale føderation, der tilknyttes Jobfunktionsroller

Denne vejledning illustrerer, hvorledes du etablerer din egen IdP med produktet [SimpleSAMLphp](#) og et egenudviklet web-baseret brugerkatalog, der fungerer som "authentication source" (på samme måde som ADFS er en overbygning til AD). Det hele er baseret på frit tilgængelige komponenter. Det er ret nemt, at tilføje og benytte egen authentication source, du kan også lave din egen.

## 1.1. Forbehold

- Der antages grundlæggende kendskab til PHP og Nginx
- Der ydes ingen support på dette produkt - du kan ikke kontakte helpdesk, hvis du har spørgsmål til SimpleSAMLphp eller brugerkataloget. Det er stillet til rådighed "As-Is".
- Dette er et praktisk eksempel, som du kan benytte som reference. Du skal tilpasse stier mm. til den platform du anvender og den måde du selv ønsker at gøre det på.
- Til denne test har vi lånt hostnavn *oiosaml-demoidp.dk* (lokalt på maskinen, defineret i *hosts*-filen) og tilhørende https-certifikat fra en tidligere test af OIOSAML. Du skal benytte eget hostnavn og egne certifikater.

## 1.2. Online Identity Providers

Forud for udarbejdelsen af denne vejledning blev det undersøgt, om det var muligt at benytte en online Identity Provider. Da man således slap for selv, at skulle etablere og konfigurere en server. Der er to grundlæggende forudsætninger et IdP-produkt skal opfylde, for anvendelse i den fælleskommunale infrastruktur:

1. Mulighed for at angive eget certifikat (FOCES)
2. Mulighed for at angive NameID format *X509SubjectName*

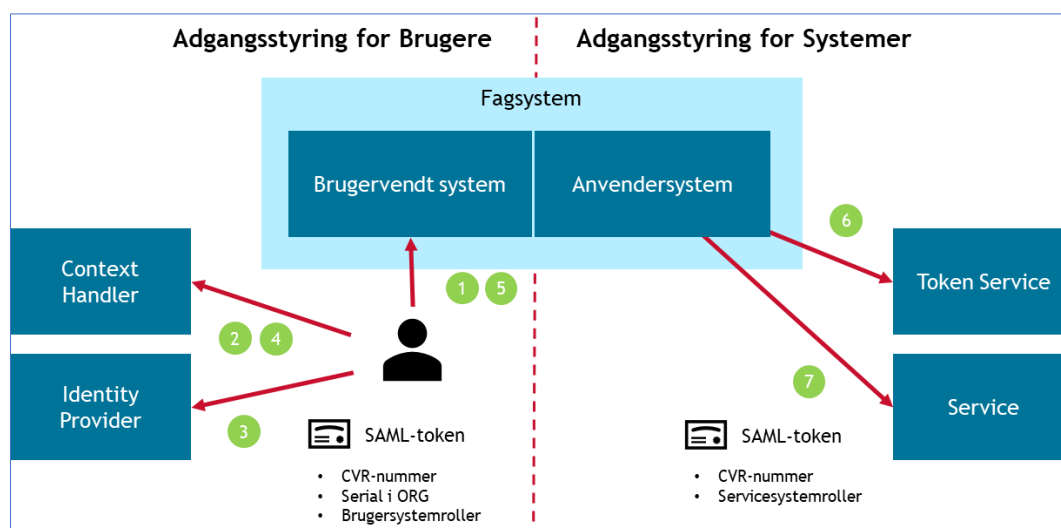
Ud af de fem undersøgte produkter var der kun ét, der opfyldte begge kriterier:

	Eget certifikat	X509SubjectName
OneLogin	Nej	Nej
Azure AD SAML	Ja	Nej
Auth0	Nej	Nej
JumpCloud	Ja	Ja
Okta	Nej	Ja

JumpCloud var eneste produkt, der opfyldte begge kriterier, men det virkede desværre ikke i praksis. Der kom en intern fejl, som deres support ikke kunne løse. Det kan tænkes, at der findes andre online produkter, der kan fungere, eller at nævnte produkter bliver opgraderet til at understøtte forudsætningerne.

## 1.3. Den fælleskommunale sikkerhedsmodel

Inden vi kaster os ud i opsætningen, lad os lige kort gennemgå samspil mellem de to sikkerhedsmodeller. Begge er baseret på SAML og tokens med bruger- eller system-roller, men de er fysisk adskilt. Det er dit ansvar, at benytte dem på korrekt vis. Typisk agerer et fagsystem både "Brugervendt system" og "Anvendersystem" og det almindelige flow er som følger:



Figur 1 - Typisk anvendelse af den fælleskommunale sikkerhedsmodel

1. Bruger tilgår fagsystemet.
2. Bruger har ingen lokal session, så bruger sendes til Context Handler (CH). Bruger vælger Identity Provider, hvor vedkommendes konto er registreret, og sendes videre til denne.
3. Bruger autentificeres af IdP og sendes tilbage med Jobfunktionsroller indlejret i bruger-token.
4. CH oversætter Jobfunktionsroller til Brugersystemroller (kun dem tilhørende det system som bruger skal sendes tilbage til).
5. Bruger kommer tilbage til fagsystemet med bruger-token indeholdende CVR-nummer for myndighed bruger tilhører, Serial-nummer henvisning til registrering af bruger i SF1500 Organisation, samt vedkommendes brugersystemroller.
6. Dernæst skal bruger fx lave et opslag i CPR-registeret fra fagsystemet. Fagsystemet trækker et system-token til service, for den myndighed bruger tilhører.
7. Fagsystemet kalder webservice med system-token på vegne af myndigheden (ikke bruger).

Du skal således selv sørge for, at kald til services fra fagsystemet foregår i samme kontekst (myndighed), som bruger der initierede handlingen.

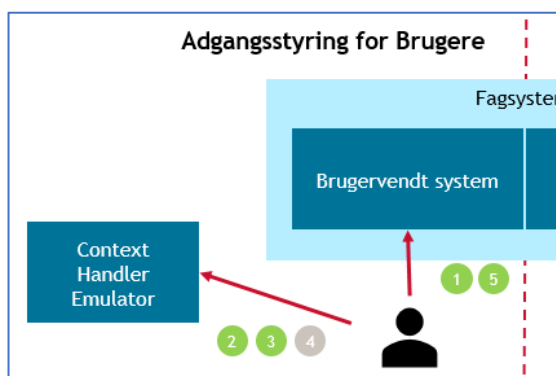
Bemærk: Det brugervendte system registrerer Context Handler som Identity Provider. Det ved ikke, at bruger sendes videre til den egentlige IdP. Dette muliggør, at vi til intern udvikling og test kan emulere Context Handler - det eneste vi skal gøre i et lokalt setup er, at sætte vores IdP til at returnere Brugersystemroller i stedet for Jobfunktionsroller.

Bemærk: Adgangsstyring for brugere foregår med client-redirect mellem komponenterne. Når du etablerer din egen IdP, så skal den blot være tilgængelig for brugerne. Den kan køre på det lokale interne netværk eller på en udviklers egen maskine, afhængigt af hvem der skal tilgå den.

Bemærk: System-tokens skal caches og genanvendes så længe de er gyldige. Så vi checker først, om vi allerede har et gyldigt token i cache i punkt 6, og benytter dette hvis det er tilfældet.

## 1.4. Lokal IdP til intern udvikling og test

Dette scenarie er relevant for alle, der skal tilslutte et brugervendt system. Du etablerer din egen lokale føderation med egen lokale IdP, og kan dernæst etablere tillid mellem denne og udvalgte udviklings- og test-instanser af dit brugervendte system. Dette lokale setup er helt isoleret og du kan oprette testbrugere for alle de myndigheder du ønsker, der er ingen begrænsninger i et lokalt setup.



Vi sætter Brugersystemroller med det samme og springer således pkt. 4 over. Set fra vores Brugervendte system fungerer vores IdP som Context Handler. Dette setup virker kun lokalt hos dig; det er kun adgangsgivende for dine egne lokale brugervendte systemer, hvortil du har etableret gensidig tillid.

## 1.5. Tilslutning af egen IdP til føderationen

Dette scenarie er som beskrevet på figuren i afsnit 1.2, hvor det er din IdP, der er registreret og meldt ind i føderationen. Du skal være registreret som myndighed med eget CVR-nummer, også kaldet *Leverandørmyndighed*.

Fordelen er, at du kan tilgå brugerflade for SF1500 Organisation, SF1510 Klassifikation samt SAPA-P på vegne af din egen myndighed. Du skal blot definere Jobfunktionsroller (JFR) i ADM, der mapper til pågældende Brugersystemroller, og dernæst tildele den eller disse JFR til dine brugere.

## 1.6. Certifikater

Du har brug for to certifikater:

### Et HTTPS-certifikat til site hvor du hoster IdP og det web-baserede brugerkatalog

Du kan benytte self-signed certifikat til dette, men det kan give udfordringer med CURL, der benyttes til at kalde det web-baserede brugerkatalogs API. Så det anbefales, at du benytter et certifikat, der er signeret af en udbredt CA. Du kan få et gratis https-certifikat hos [Letsencrypt](#).

### Et certifikat til signering af SAML-beskeder

Du kan også benytte et self-signed certifikat til SAML-signering, hvis du kører det lokale setup. Men hvis din IdP skal tilmeldes føderationen, så er det et krav, at du anvender et funktionscertifikat (FOCES). Hvis dit fagsystem allerede er registreret som anvendersystem, og du allerede har et funktionscertifikat til dette, da går det fint at benytte samme til din IdP. MEN - du kan ikke registrere et Brugervendt system og Identity Provider med samme certifikat.

### Eksport af offentlig version og privat nøgle

Både Nginx og SimpleSAMLphp kræver, at man angiver certifikatet med den offentlige del og den private nøgle separat. Hvis dit certifikat er leveret i pfx/p12 format, da skal du eksportere delene hver for sig. Dette gjorde vi med [OpenSSL](#).

#### Eksport af offentlige del:

```
C:\OpenSSL\x64\bin>openssl.exe pkcs12 -in <fuld-sti>.pfx -out <fuld-sti>.cer -nokeys -password pass:<kode>
```

#### Ekport af private nøgle:

```
C:\OpenSSL\x64\bin>openssl.exe pkcs12 -in <fuld-sti>.pfx -out <fuld-sti>.priv.key -nodes -nocerts -password pass:<kode>
```

Der skal anførselstegn rundt om filnavne, hvis de indeholder mellemrum. Parameter -nodes angiver, at den eksporterede private nøgle ikke skal krypteres med en adgangskode.

## 1.7. Kort om roller

Vi som leverandør starter med at definere de nødvendige Brugersystemroller (BSR) i vores applikation, fx "Sagsbehandler", "Administrator" og "Supporter". Vi registrerer dem dernæst på vores Brugervendte system i den fælleskommunale administration:



Brugersystemroller		
		<a href="#">+ Opret brugersystemrolle</a>
UUID	Navn <sup>^</sup>	Dataafgrænsningstyper
<input type="text"/>	<input type="text"/>	<input type="text"/>
4e4...	Administrator	
17e...	Sagsbehandler	
694...	Supporter	

Hver myndighed, der skal benytte vores applikation, definerer dernæst selv én eller flere Jobfunktionsroller, i henhold til hvorledes de selv ønsker, at administrere deres brugeres rettigheder. I dette eksempel har vi som Korsbæk kommune oprettet en Jobfunktionsrolle, der indeholder to BSR:

KDI Testrolle <small>(På vegne af Korsbaek Kommune)</small>		
Navn:	KDI Testrolle	UUID:
Entityld:	<a href="http://korsbaek.dk/roles/jobrole/testrolle/1">http://korsbaek.dk/roles/jobrole/testrolle/1</a>	Oprett
Beskrivelse:	Til vejledning "Adgangsstyring for brugere - Egen Identity Provider"	Ændre
Delegeret til:		
Brugersystemroller	<b>System <sup>^</sup></b>	<b>Rolle</b>
	KDI Brugervendt system Demo	Sagsbehandler
	KDI Brugervendt system Demo	Supporter

Når vi i vores Identity Provider tildeler en bruger pågældende JFR:

- <http://korsbaek.dk/roles/jobrole/testrolle/1>

Da skal vi sørge for, at vores Identity Provider medsender denne JFR i det SAML-token som bruger sendes tilbage til Context Handler med (i attribut med navn "Privileges\_intermediate"). Context Handler vil dernæst oversætte denne JFR til disse to BSR, inden bruger sendes videre til vores fagsystem:

- <http://test.mit-fagsystem.dk/roles/usersystemrole/sagsbehandler/1>
- <http://test.mit-fagsystem.dk/roles/usersystemrole/supporter/1>

Når bruger kommer til vores fagsystem med SAML-token, da aflæser vi tildelte roller og giver vedkommende adgang derefter. I praksis ser det ud som følger, når vi base64-afkoder Privileges\_intermediate:

```
<bpp:PrivilegeList xmlns:bpp="http://itst.dk/oiosaml/basic_privilege_profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PrivilegeGroup Scope="urn:dk:gov:saml:cvrNumberIdentifiser:11111111">
    <Privilege>http://test.mit-
fagsystem.dk/roles/usersystemrole/sagsbehandler/1</Privilege>
  </PrivilegeGroup>
  <PrivilegeGroup Scope="urn:dk:gov:saml:cvrNumberIdentifiser:11111111">
    <Privilege>http://test.mit-fagsystem.dk/roles/usersystemrole/supporter/1</Privilege>
  </PrivilegeGroup>
</bpp:PrivilegeList>
```

## 2. PHP OG NGINX

### 2.1. PHP

Her blev benyttet [PHP version 7.4.29](#), da SimpleSAMLphp ikke er kompatibel med nyere versioner af PHP (de skriver på deres hjemmeside, at de arbejder på det). Vi hentede pakken til Windows og udpakkede filerne i folder C:\PHP.

Filen *php.ini-development* blev kopieret til *php.ini* og følgende extensions blev aktiveret:

C:\PHP\php.ini

```
extension=curl
extension=intl
extension=mbstring
extension=openssl
```

Der er forskellige måder at aktivere PHP-engine på og gøre tilgængelig for webserver, afhængigt af platform man anvender. I denne test benyttede vi PHP CGI og startede den manuelt:

```
C:\PHP>php-cgi -b 127.0.0.1:9000
```

### 2.2. Nginx

Vi benytter [Nginx](#) i egenskab af Webserver og seneste version blev hentet og udpakket i folder C:\Nginx. Her gengivet de centrale dele af konfigurationen:

C:\Nginx\conf\nginx.conf

## Konfiguration af HTTPS

```
server {
    listen 8090 ssl;
    server_name ...;
    root c:/Nginx/html;

    ssl_certificate      C:\<https-certifikat>.cer;
    ssl_certificate_key  C:\<https-certifikat>.priv.key;
    ssl_protocols        TLSv1.2 TLSv1.3;
    ...
}
```

## Aktivering af PHP

```
location ~ /\.php$ {
    include fastcgi_params;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}
```

## SimpleSAMLphp konfiguration

```
location ^~ /saml {
    alias c:/Nginx/html/saml;
    location ~^(?<prefix>/saml)(?<phpfile>.+?.php)(?<pathinfo>/.*)?$ {
        include fastcgi_params;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $request_filename;
        fastcgi_param SCRIPT_NAME /saml$phpfile;
        fastcgi_param PATH_INFO $pathinfo if_not_empty;
    }
}
```

Vi valgte navnet "saml" til folderen, hvortil vi kopierede www-delen af SimpleSAMLphp koden. Ovenstående konfiguration gav lidt hovedpine og krævede søgning på nettet. Hvis routing fejler i SimpleSAMLphp, så tjek denne konfiguration. Til denne test startede vi Nginx manuelt:

```
c:\Nginx>nginx
```

## 3. SIMPLESAMLPHP

### 3.1. Installation

Seneste version af [SimpleSAMLphp](#) blev hentet og udpakket til c:\simplesamlphp.

Indholdet af folder C:\simplesamlphp\www\ blev kopieret til c:\nginx\html\saml\.

### 3.2. Certifikater

#### C:\simplesamlphp\cert

Her gemmer vi offentligt certifikat + privat nøgle, der skal benyttes af vores IdP til signering af SAML-beskeder (skal være FOCES hvis IdP skal tilsluttes den fælleskommunale føderation):

- <dit-saml-signing-certifikat>.pem
- <dit-saml-signing-certifikat>.priv-key.pem

Offentlig del og privat nøgle gemt separat, som nævnt i afsnit 1.6 Certifikater. Her skal vi også senere gemme certifikater fra service providers, der skal tilsluttes vores IdP.

### 3.3. \_include.php

#### C:\Nginx\html\saml\\_include.php

Fortæl web-projektet hvor du har placeret SimpleSAMLphp filerne:

```
$root_folder = "C:/simplesamlphp";
```

### 3.4. Config.php

Her er gengivet de centrale konfigurationsparametre. Du skal angive din egen værdi alle steder markeret med gult, og kan selvfølgelig ændre generelt som du ønsker. Se hjælpeteksten i filen for detaljer.

#### C:\simplesamlphp\config\config.php

Parameter	Værdi
baseurlpath	<a href="https://oiosaml-demoidp.dk:8090/saml/">https://oiosaml-demoidp.dk:8090/saml/</a>
secretsalt	entilfældignøgle
auth.adminpassword	enhemmeligkode
debug	Her kan du aktivere logning af SAML-beskeder (se hjælpetekst i filen)
showerrors	true Nyttigt i forbindelse med opsætning, kan sættes til "false" senere
logging.level	'Warning'
enable.saml20-idp	true
module.enable	'saml' => true

session.duration	60 Sat til lav værdi mens vi tester
language.available	'en' For at simplificere brugerfladen
language.default	'en'
production	false Så længe vi tester opsætningen
authproc.idp	//99 => 'core:LanguageAdaptor' vi udkommenterede dette filter, da det tilføjede en uønsket "preferredLanguage" attribut i assertion

### 3.5. saml20-idp-hosted.php

C:\simplesamlphp\metadata\saml20-idp-hosted.php

Dokumentation: <https://simplesamlphp.org/docs/latest/simplesamlphp-reference-idp-hosted.html>

```
$metadata['DYNAMIC:1'] = [
    'host' => '__DEFAULT__',
    'privatekey' => 'KOMBIT AS - KDI STS SFTP IBA Test2.priv-key.pem',
    'certificate' => 'KOMBIT AS - KDI STS SFTP IBA Test2.pem',
    'auth' => 'kdi-authentication',
    'authproc' => [
        1 => [
            'class' => 'saml:AttributeNameID',
            'attribute' => 'nameId',
            'Format' => 'urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName',
            'SPNameQualifier' => false
        ],
    ],
    'attributes.NameFormat' => 'urn:oasis:names:tc:SAML:2.0:attrname-format:basic',
    'NameIDFormat' => 'urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName',
    'saml20.sign.assertion' => true,
    'saml20.sign.response' => false,
    'redirect.sign' => true,
    'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    'attributes' => [
        'dk:gov:saml:attribute:CvrNumberIdentifier',
        'dk:gov:saml:attribute:KombitSpecVer',
        'dk:gov:saml:attribute:SpecVer',
        'dk:gov:saml:attribute:AssuranceLevel',
        'dk:gov:saml:attribute:Privileges_intermediate'
    ]
];
```

- \$metadata-nøglen er EntityID for IdP. Som udgangspunkt er den automatisk ud fra host-navn, men du kan angive din egen (se dokumentationen).
- Du skal ændre “privatekey” og “certificate” til dit eget certifikat
- “auth” peger på vores egen authentication source, som vi opretter i et efterfølgende afsnit
- I “authproc” parameter har vi angivet, at værdien til Assertion\Subject\NameID skal aflæses fra attributten “nameid” i data, der returneres fra vores authentication source.

### 3.6. authsources.php

#### C:\simplesamlphp\config\authsources.php

I starten af filen har vi tilføjet vores egen authentication source “kdi-authentication” med reference til det modul og den klasse, der håndterer autentificering mod vores brugerkatalog.

```
$config = [  
    'kdi-authentication' => ['kdiauth:KDIAuthAPI'],  
    ...
```

Det er rigtigt nemt, at definere og implementere egen authentication source. Du behøver ikke benytte det brugerkatalog vi har implementeret i denne guide. Du kan også lave en simpel version, der fx indlæser brugere fra en fil. Man skal blot implementere et *login(\$username, \$password)* interface, som beskrevet i det efterfølgende afsnit.

### 3.7. KDIAuthAPI.php

Dokumentation: <https://simplesamlphp.org/docs/latest/simplesamlphp-customauth.html>

Dette er vores authentication source, som vi har kaldt kdiauth:KDIAuthAPI (module:class). Den laver et webkald til API i vores brugerkatalog med brugernavn/adgangskode, og der returneres information om brugeren ved succes. Vi har gjort følgende:

- Oprettet folder “C:\simplesamlphp\modules\kdiauth\”
- Kopieret alt indhold fra “..\modules\exampleauth\” til “..\modules\kdiauth\”
- Slettet alle filer i “..\modules\kdiauth\lib\Auth\Source\”
- Oprettet en ny fil “KDIAuthAPI.php” i “..\modules\kdiauth\lib\Auth\Source\”

#### C:\simplesamlphp\modules\kdiauth\lib\Auth\Source\KDIAuthAPI.php

```
<?php  
namespace SimpleSAML\Module\kdiauth\Auth\Source;  
  
class KDIAuthAPI extends \SimpleSAML\Module\core\Auth\UserPassBase  
{  
    protected function login($username, $password) {
```

```
$API_URL = 'https://mit-brugerkatalog.dk/api.php';
$API_KEY = 'dd64ed19-60ec-467b-ad83-017f695b1598';
$postData = json_encode(['username' => $username, 'password' => $password]);

$ch = curl_init($API_URL);
curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => ['Content-Type: application/json', 'api-key: '.$API_KEY],
    CURLOPT_POST => true,
    CURLOPT_POSTFIELDS => $postData,
    CURLOPT_SSL_VERIFYPEER => true,
    CURLOPT_CAINFO => 'C:\PHP\cacert.pem' // kan også sættes i php.ini
]);

$result = curl_exec($ch);

if ($result === false) {
    $error_msg = curl_error($ch); // Tilføj logging
    curl_close($ch);
    throw new \SimpleSAML\Error\Error('WRONGUSERPASS');
}

$user = json_decode($result, true);
curl_close($ch);

if (isset($user['error'])) throw new \SimpleSAML\Error\Error('WRONGUSERPASS');

return [
    'nameId' => ['C=DK,O='.$user['CVR'].'.CN='.$user['Name'].',Serial='.$user['Serial']],
    'dk:gov:saml:attribute:CvrNumberIdentifier' => [$user['CVR']],
    'dk:gov:saml:attribute:KombitSpecVer' => ['1.0'],
    'dk:gov:saml:attribute:SpecVer' => ['DK-SAML-2.0'],
    'dk:gov:saml:attribute:AssuranceLevel' => ['3'],
    'dk:gov:saml:attribute:Privileges_intermediate' => [$user['PrivilegesIntermediate']]
];
}
}
?>
```

API-nøglen vælger du selv - husk at sætte den samme når du konfigurerer det web-baserede brugerkatalog.

Huske at sørge for, at Root CA for dit brugerkatalog HTTPS-certifikat er med i cacert.pem. Ellers vil CURL-kald til api.php ved autentificering fejle.

Det var i saml20-idp-hosted.php parameter "authproc", at vi definerede, at Subject->NameID for en bruger skal aflæses fra array-parameter 'nameId'.

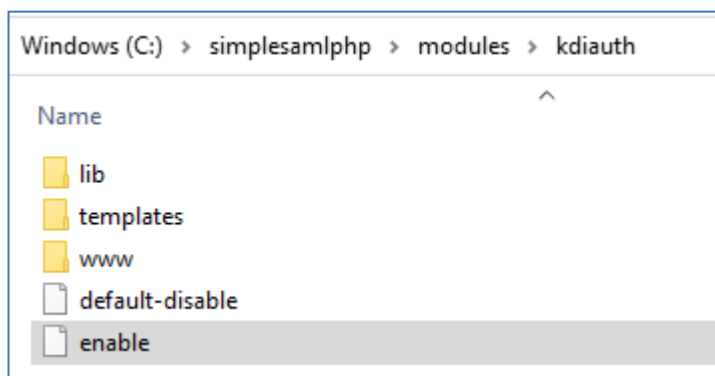
NameID format X509SubjectName er (eksempel):

```
C=DK,O=11111111,CN=Bruce Lee,Serial=fd2ed2a9-09fc-4b4f-98a6-6d7bf206d088
```

Hvor O (Organisation) = CVR-nummer for myndighed, CN (Common Name) er brugers navn i IdP (i praksis ikke nødvendigvis brugers fulde korrekte navn), og Serial er nøglen til brugers entitet i SF1500 Organisation.

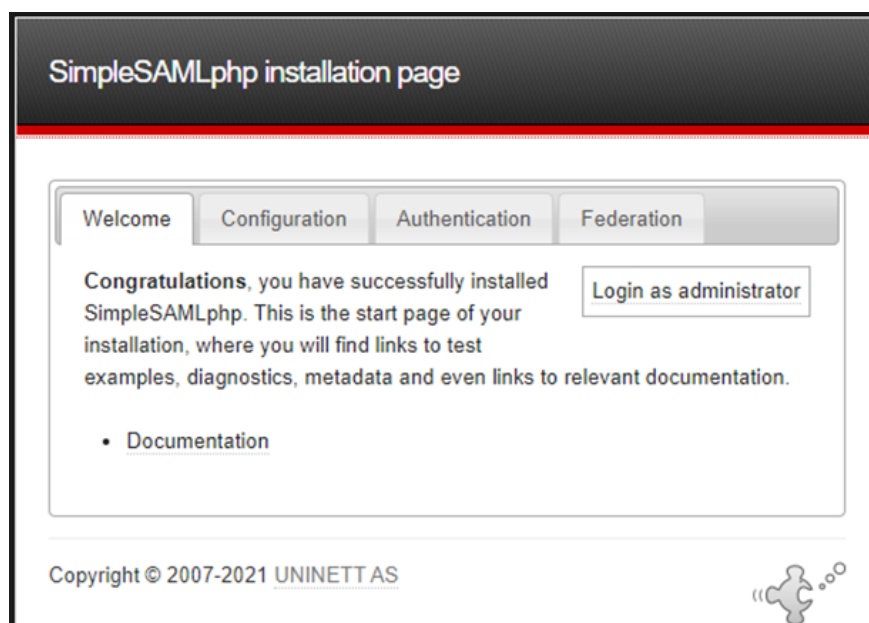
Nye brugere skal således oprettes i Organisation og få tildelt denne nøgle. I praksis benytter de fleste kommuner AD som brugerkatalog og har automatisk synkronisering mellem dette og Organisation. I testøjemed kan du manuelt oprette brugere i Organisation for din leverandørmyndighed. Bemærk - det er tiltænkt at fagsystemet, efter bruger-login, efterfølgende laver opslag i Organisation og henter mere information om brugeren. Hvis du ikke har brug for dette i dit fagsystem, da kan du angive en tilfældig nøgle for hver bruger.

Vi aktiverer modulet ved at oprette en tekstfil med navn "enable" og indhold "enable" i roden af folderen:



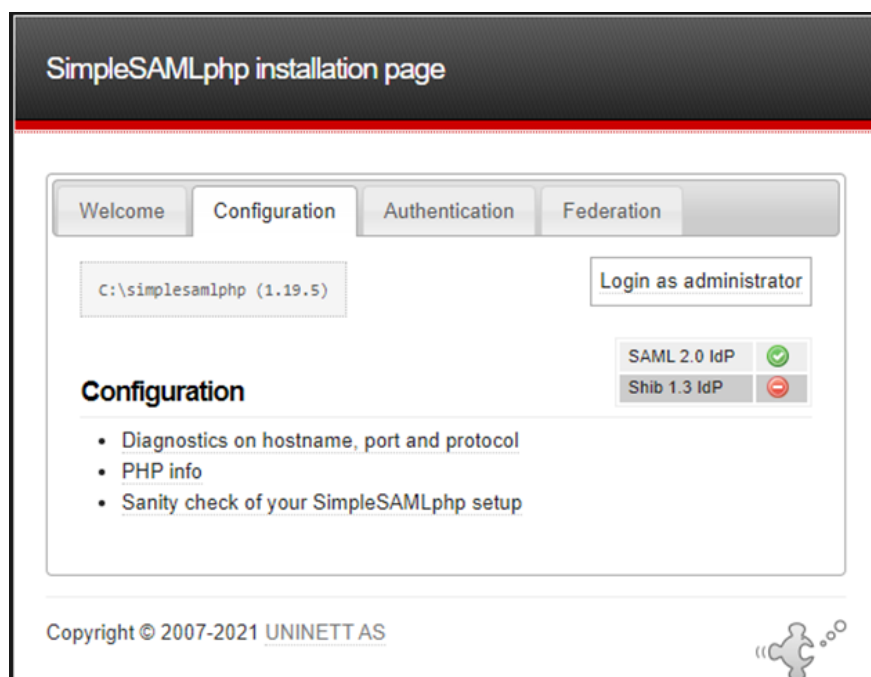
### 3.8. Opsætning tjek

Nu er opsætningen klar og vi kan teste, at sitet kører:

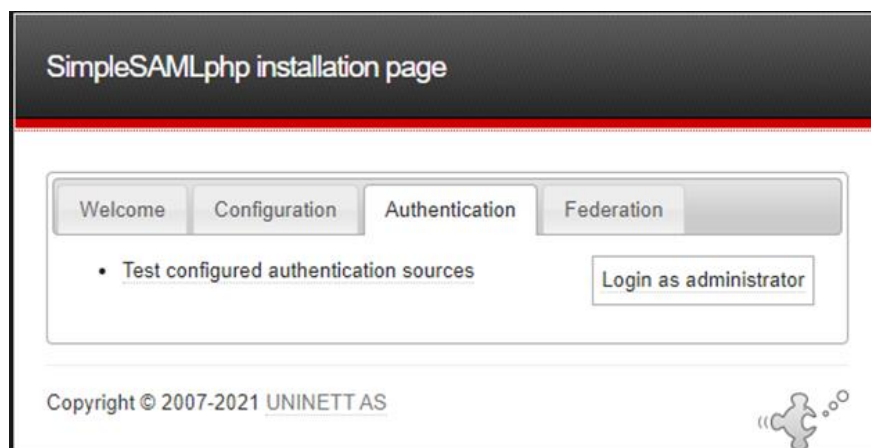




Vælg "Configuration -> Sanity check" så du kan se, at alt er på plads.



"Authentication -> Test configured authentication sources" skal vi bruge senere, når vi har etableret vores brugerkatalog.



Under "Federation" finder du metadata for din IdP.

Bemærk: Mht. metadata for din IdP, som du skal registrere i FK ADM, hvis din IdP skal tilsluttes den fælleskommunale føderation - den genereres uden blokken `<KeyDescriptor use="encryption">`, da vi (i hvert fald i testøjemed) ikke ønsker at vores Assertion krypteres. Men denne blok er påkrævet ved validering af metadata-filen i FK ADM. Workaround var at tage en kopi `<KeyDescriptor use="signing">` blokken, placere den nedenunder, og dernæst omdøbe "signing" til "encryption" i den nye blok:

## Før

```
<EntityDescriptor ...>
  <IDPSSODescriptor ...>
    <KeyDescriptor use="signing">
      ...
    </KeyDescriptor>
    ...
  </IDPSSODescriptor>
</EntityDescriptor>
```

## Efter

```
<EntityDescriptor ...>
  <IDPSSODescriptor ...>
    <KeyDescriptor use="signing">
      ...
    </KeyDescriptor>
    <KeyDescriptor use="encryption">
      ...
    </KeyDescriptor>
    ...
  </IDPSSODescriptor>
</EntityDescriptor>
```

Yderligere, så kræver Context Handler, at der også er en HTTP-POST binding med i metadata, og denne er ikke med i den metadata SimpleSAMLphp genererer. Så vi har tilføjet den manuelt:

## Før

```
<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName</NameIDFormat>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SSOService.php"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SingleLogoutService.php"/>
</IDPSSODescriptor>
```

## Efter

```
<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName</NameIDFormat>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SSOService.php"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SingleLogoutService.php"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SSOService.php"/>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://oiosaml-demoidp.dk:8090/saml/saml2/idp/SingleLogoutService.php"/>
</IDPSSODescriptor>
```

## 4. BRUGERKATALOG

### 4.1. Installation

Dette er et egetudviklet brugerkatalog, der stilles til rådighed As-Is. Der ydes ingen support og du er velkommen til at tilpasse det, så meget du ønsker. Du henter filen [her](#).

Produktet understøtter flere myndigheder, i tilfælde af, at du vil benytte det til et lokalt setup. Produktet har nogle begrænsninger, som du er velkommen til at udbedre:

- Man kan ikke slette entiteter
- Autentificering logges ikke
- Dataafgrænsning parametre gælder på tværs af myndigheder
- Man kan ikke tilføje egne attributter, og dermed kan der kun sættes dataafgrænsning på roller
- Der kan kun sættes én dataafgrænsningsværdi på en rolle

### 4.2. Config.php

Du skal opdatere følgende parametre:

DATA_FOLDER	Opret en folder til opbevaring af data. Fx "c:\IdP-Brugerkatalog". Kontekst (bruger) som web-applikationen kører i skal have læse/skrive-adgang til folder. Hvis du ønsker, at starte forfra med en ren installation, da kan du blot slette alt indhold i denne folder og køre setup.php igen.
API_KEY	En selvvalgt nøgle - fx et UUID. Vores authentication modul i SimpleSAMLphp kalder med denne nøgle i http-header, så husk også at opdatere der (afsnit 3.7).

### 4.3. Setup.php

Dette script opretter en myndighed og nogle eksempel roller/brugere, samt en administrativ bruger. Husk at gå hele script igennem og ændre/tilføje CVR-nummer, adgangskoder, osv. inden du kører det.

Du skal udkommentere den første linje i scriptet, for at kunne køre det. Du kører scriptet ved at kalde det i browser: <https://<dit-brugerkatalog>/setup.php>. Efter du har kørt setup, husk at aktivere den første linje igen. Så du ikke risikerer, at få data lagt ind dobbelt.

Hvis du har behov for at starte forfra, da kan du blot slette alt indhold i data-folder og køre setup-script igen.

## 4.4. Roller og brugere

Nu har du et web-baseret brugerkatalog, hvor du kan oprette roller og brugere, til benyttelse fra vores egen Identity Provider. Her eksempel:

The screenshot shows a web interface for 'IdP Brugerkatalog'. At the top, there is a navigation bar with 'IdP Brugerkatalog', a 'Menu' dropdown, 'Korsbæk Kommune', and a 'Logud' button. Below the navigation bar, the main heading is 'IdP Brugere'. There are three tabs: 'Liste', 'Opret ny', and 'Bruger #1'. The 'Bruger #1' tab is active, showing a form with the following fields:

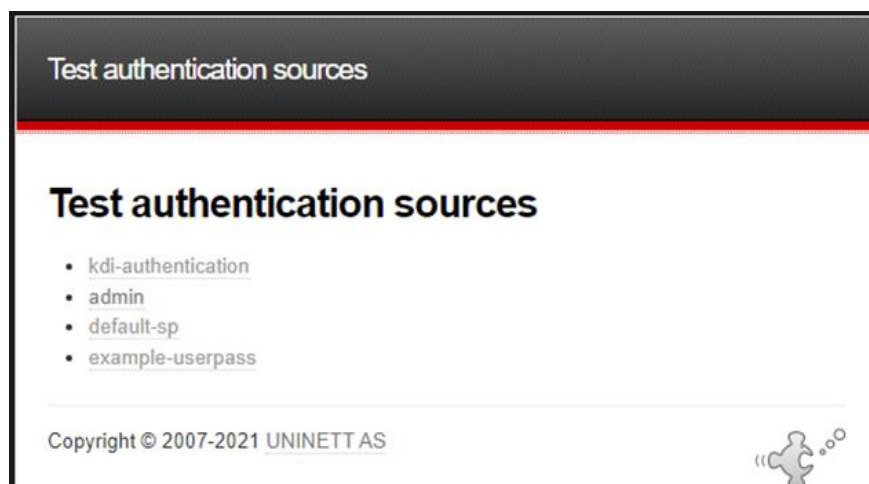
- Navn: Test Testesen
- Serial: fd2ed2a9-09fc-4b4f-98a6-6d7bf206d088
- Brugernavn: testbruger01
- Adgangskode: Test1234!
- Rolle:  http://korsbaek.dk/roles/jobrole/KlassifikationLaesKorsbaek/1

At the bottom of the form is a blue button labeled 'Opdater' with a downward arrow icon.

Nu er vi klar til at teste det komplette setup af vores Identity Provider + Brugerkatalog.

## 5. TEST AF IDP OPSÆTNING

I SimpleSAMLphp brugerfladen, vælg "Authentication -> Test authentication sources":




Vælg "kdi-authentication" (eller det navn du selv har givet det). Indtast dernæst brugernavn/adgangskode for en bruger fra brugerkataloget:

Enter your username and password


---

### Enter your username and password

A service has requested you to authenticate yourself. Please enter your username and password in the form below.



Username



Password

SimpleSAMLphp vil dernæst kalde api.php i vores brugerkatalog med brugernavn og adgangskode, og dernæst benytte den returnerede data til at populere de attributter, der vil blive sat i et SAML-token for brugeren:

SAML 2.0 SP Demo Example

---

### SAML 2.0 SP Demo Example

Hi, this is the status page of SimpleSAMLphp. Here you can see if your session is timed out, how long it lasts until it times out and all the attributes that are attached to your session.

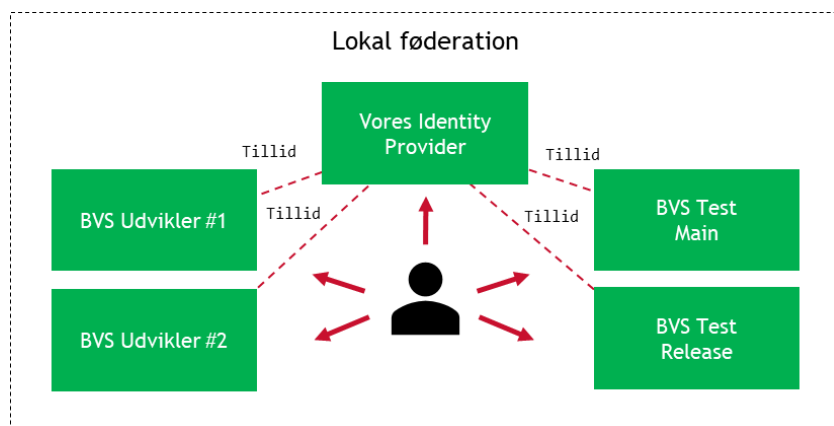
#### Your attributes

nameId	C=DK,O=11111111,CN=Test Testesen,Serial=fd2ed2a9-09fc-4b4f-98a6-6d7bf206d088
dk:gov:saml:attribute:CvrNumberIdentifier	11111111
dk:gov:saml:attribute:KombitSpecVer	1.0
dk:gov:saml:attribute:SpecVer	DK-SAML-2.0
dk:gov:saml:attribute:AssuranceLevel	3
dk:gov:saml:attribute:Privileges_intermediate	PFByaXZpbGVnZUxpc3QgeG1sbnM9Imh0dHA6Ly9pdHN0LmRlL29pb3NhbWwvYmEz

## 6. ANVENDELSE I LOKAL FØDERATION

### 6.1. Introduktion

Som beskrevet i introduktionen, så kan det lokale setup benyttes til frit at teste din sikkerhedsmodel på udviklings- og interne test-maskiner. Du etablerer en lokal IdP og etablerer trust mellem denne og de Brugervendte systemer (BVS) du skal teste fra, ved at udveksle metadata.



Du kan selv definere myndigheder, brugersystemroller og test-brugere i brugerkataloget efter behov.

I dette eksempel har vi etableret et brugervendt system med OIOSAML .NET eksempelkoden, som beskrevet i [kom-godt-i-gang Tilslut Brugervendt system](#).

### 6.2. Registrering af Service Provider i IdP

Vores Brugervendte system (BVS) er *Service Provider* (SP) i SAML-terminologi. Vi etablerer tillid til vores BVS ved at registrere det i følgende konfigurationsfil:

**C:\simplesamlphp\metadata\saml20-sp-remote.php**

```
$metadata[ 'https://test.bvs-main.dk' ] = [
    'name' => 'Mit fagsystem',
    'AssertionConsumerService' => 'https://localhost:20002/login.ashx',
    'certificate' => '<dit-bvs-certifikat>.pem',
    'NameIDFormat' => 'urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName',
    'saml20.sign.assertion' => true,
    'saml20.sign.response' => false,
    'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    'SingleLogoutService' => 'https://localhost:20002/logout.ashx'
];

$metadata[ 'https://test.bvs-release.dk' ] = [
    ...
];
```

Du angiver Entity ID for SP som nøgle i \$metadata array.

Bemærk: I det lokale setup kan du måske benytte samme certifikat til alle parterne (er ikke testet). Certifikater gemmes i SimpSAMLphp "cert" folder - den offentlige version i PEM-format. I den fælleskommunale føderation skal alle parter (SP + IdP) have eget unikke certifikat.

Bemærk: vi har ikke registreret SP metadata-filen, som man ellers gør i mange rammeværk. Vi har aflæst de essentielle parametre fra dens metadata og tilføjet dem til konfigurationsfilen manuelt.

### 6.3. Registrering af IdP i Service Provider

De forskellige rammeværk har hver deres måde, hvorpå man registrerer IdP, man vil etablere tillid til. Med OIOSAML .NET eksempelkode, som vi har benyttet til denne test, gøres dette ved at gemme IdP metadata-fil i folder "src\dk.nita.saml20\WebsiteDemo\idp-metadata". Filnavnet har ingen betydning - der må blot ikke være mellemrum.

### 6.4. Test

Vi har etableret gensidig tillid, og nu kan vi teste vores setup. Vi starter vores OIOSAML brugervendte system:

**OIOSAML.NET**

---

- [Go to My Page.](#)
- [Go to My Page \(require high assurance level\)](#)
- [Go to My Page \(request a Professional attribute profile\)](#)

**Metadata**

The identity provider and the service provider must exchange metadata in order to establish SAML connections. The Identity provider's metadata should be put in the directory  
"C:\Users\xmag\source\repos\OIOSAML3\src\dk.nita.saml20\WebsiteDemo\idp-metadata".

The metadata of the service provider can be downloaded [here](#).

---

© OIOSAML.NET ([www.oiosaml.info](http://www.oiosaml.info)).

Vi vælger "Go to My Page" og bliver dernæst bedt om at vælge IdP vi vil autentificere mod. Vi vælger vores egen IdP (som her er nederst på listen, de andre kommer fra OIOSAML-projektet):

**Choose Identity Provider**

Please choose the identity provider of your choice from the list below:

<https://saml.test-devtest4-nemlog-in.dk>  
<https://oiosaml-demoidp.dk:20001/>  
<https://oiosaml-demoidp.dk:8090/saml/saml2/idp/metadata.php>





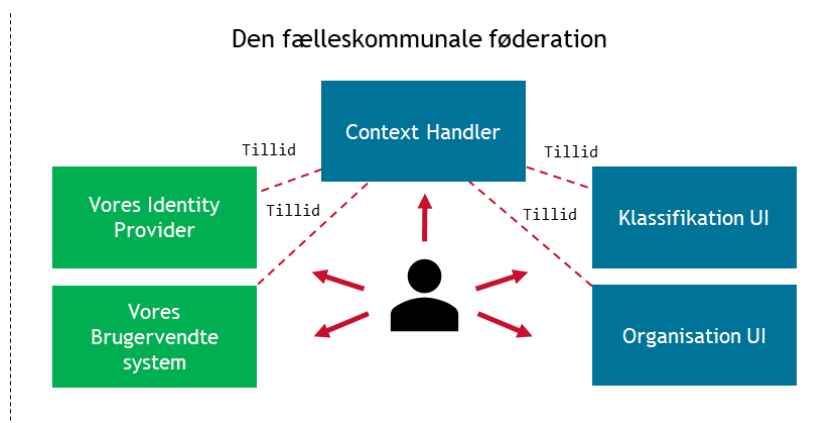


## 7. TILSLUTTET DEN FÆLLESKOMMUNALE FØDERATION

### 7.1. Introduktion

Denne mulighed er relevant for leverandører, der i ExtTest er oprettet som egen myndighed i testøjemed - også kaldet *Leverandørmyndighed*. Når ens egen IdP tilsluttes føderationen, da kan den også anvendes til at give adgang til Klassifikation, Organisation eller SAPA-P på vegne af ens egen myndighed.

SAML er baseret på gensidig tillid og Context Handler (CH) er den centrale komponent. Hver part i føderationen har etableret tillid til CH og vice versa - de kender kun CH og ikke de andre parter. Eksempel: CH stoler på din IdP og Klassifikation stoler på CH. Dermed vil Klassifikation acceptere et bruger-token udstedt af CH på baggrund af et bruger-token udstedt af din IdP.



Det vi skal gøre er at:

- I det fælleskommunale administrationsmodul
  - Registrere vores IdP med metadata
  - Registrere føderationsaftale der tillader, at vores IdP kan autentificere brugere fra vores myndighed
  - Definere Jobfunktionsroller og tilføj de Brugersystemroller der skal knyttes til hver
- I vores Identity Provider
  - Registrere Context Handler metadata
  - Definere Jobfunktionsroller så de kan tilknyttes brugere

### 7.2. Registrering af IdP i føderationen

Du kan angive flere formål, når du registrerer et IT-system i det fælleskommunale administrationsmodul (FK ADM). Det er fx typisk og giver god mening, at registrere et IT-system som både Anvendelsesystem (skal kalde services) og Brugervendt system (skal benytte adgangsstyring for brugere). Når det gælder

registrering af Identity Provider, da anbefales det, at du registrerer denne som et separat IT-system. Da det har en isoleret og specifik opgave.

NB: Vi måtte tilføje encryption-blok og HTTP-POST bindings manuelt i metadata-filen, for at gøre den kompatibel med Context Handler - se afsnit 3.8 Opsætning tjek.

Her har vi registreret vores "KDI Identity Provider Demo" IT-system med metadata. Vores IdP er dermed tilsluttet den fælleskommunale føderation.

Der går et par minutter efter du har gemt, og så vil registreringen være provisioneret til Context Handler. Når du dernæst vælger login fra fx [Demo Brugervendt System](#), da vil den nyligt oprettede IdP kunne vælges som "authentication method":

Inden du kan benytte den til at autentificere brugere fra din egen myndighed, skal der oprettes en føderationsaftale.

### 7.3. Registrering af føderationsaftale

Som *Leverandørmyndighed* kan du selv anmode om føderationsaftale til din IdP, på vegne af din egen myndighed, og godkende den. Du vælger blot "Føderationsaftaler i venstre menu i den fælleskommunale administration og følger vejledning.

En føderationsaftale ser ud som følger:

KDI Identity Provider Demo	
Status:	Godkendt
System:	KDI Identity Provider Demo
Begrundelse:	Til test af simplesamlphp
Betingelser:	<a href="#">Vis vilkår og betingelser ved anmodning</a>
Myndighed:	Korsbaek Kommune
Kommentar:	Godkendt til test

Bemærk: At du bør ikke anmode om føderationsaftale til din IdP for andre myndigheder end din egen.

### 7.4. Registrering af Jobfunktionsrolle

Vi har registreret en "testrolle1" Jobfunktionsrolle for vores myndighed og tilknyttet to Brugersystemroller i vores eget Brugervendte system.

## Opret jobfunktionsrolle (På vegne af Korsbæk Kommune)

Navn: \*

**EntityId** ⓘ

Rollenavn: \*

<http://korsbaek.dk/roles/jobrole/testrolle/1>

Beskrivelse:

Delegeret til: [Klik her for at uddelegere jobfunktionsrollen](#)

[+ Tilknyt brugersystemroller](#)

Brugersystemroller	System ^	Rolle	Dataafgrænsning
	KDI CTT Test System #2	Sagsbehandler	
	KDI CTT Test System #2	Supporter	

[Gem](#) [Annuller](#)

Den komplette parameter er: "http://korsbaek.dk/roles/jobrole/testrolle/1". Vi opretter den samme rolle i vores brugerkatalog:

IdP Brugerkatalog Menu Korsbæk Kommune Logud

### Roller

[Liste](#) [Opret ny](#)

EntityId

Dataafgrænsning  Følsomhed  KLE

[+ Opret](#)

Vi kan dernæst tildele rollen til en test-bruger:

IdP Bruger katalog Menu Korsbæk Kommune Logud

---

## IdP Brugere

Liste Opret ny Bruger #1

Navn	<input type="text" value="Test Testesen"/>
Serial	<input type="text" value="fd2ed2a9-09fc-4b4f-98a6-6d7bf206d088"/>
Brugernavn	<input type="text" value="testbruger01"/>
Adgangskode	<input type="text" value="Test1234!"/>
Rolle	<input checked="" type="checkbox"/> <a href="http://korsbaek.dk/roles/jobrole/KlassifikationLaesKorsbaek/1">http://korsbaek.dk/roles/jobrole/KlassifikationLaesKorsbaek/1</a>
Rolle	<input checked="" type="checkbox"/> <a href="http://korsbaek.dk/roles/jobrole/testrolle/1">http://korsbaek.dk/roles/jobrole/testrolle/1</a>

På samme vis har vi oprettet og tildelt en jobfunktionsrolle, der i det fælleskommunale administrationsmodel er konfigureret til at give læse-adgang til Klassifikation.

## 7.5. Registrering af Context Handler i IdP

Du finder metadata for Context Handler (CH) på informationssiden for [Adgangsstyring for brugere](#). Vi har aflæst værdierne derfra og tilføjet i konfigurationsfilen:

C:\simplesamlphp\metadata\saml20-sp-remote.php

```
$metadata[ 'https://saml.adgangsstyring.eksterntest-stoettesystemerne.dk' ] = [
    'name' => 'Context Handler ExtTest',
    'AssertionConsumerService' => 'https://adgangsstyring.eksterntest-
stoettesystemerne.dk/runtime/saml2auth/consume.idp',
    'certificate' => 'context-handler.pem',
    'NameIDFormat' => 'urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName',
    'saml20.sign.assertion' => true,
    'saml20.sign.response' => false,
    'signature.algorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha256',
    'SingleLogoutService' => 'https://adgangsstyring.eksterntest-
stoettesystemerne.dk/runtime/saml2/issue.idp'
];
```

Nøgler i \$metadata er entityId for CH. Parameter "name" er blot til visning.

Certifikat benyttet af Context Handler har “*adgangsstyring*” i navnet og du henter det på [Certifikater](#)-siden. Vi har hentet certifikater-pakken for ExtTest og gemt *adgangsstyring* certifikatet som:

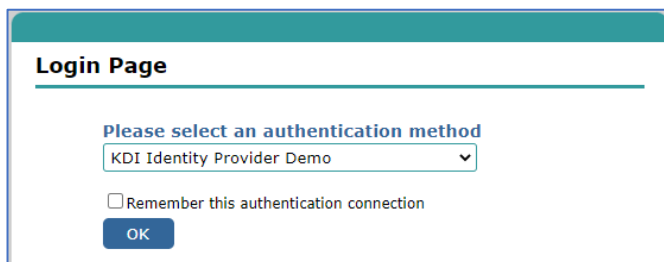
**C:\simplesamlphp\cert\context-handler.pem**

Bemærk: Vi har registreret Context Handler som en Service Provider. Det er den jo ikke, men set fra vores IdP fungerer den således - brugere kommer derfra med SAML-request og de skal sendes tilbage dertil med SAML-response.

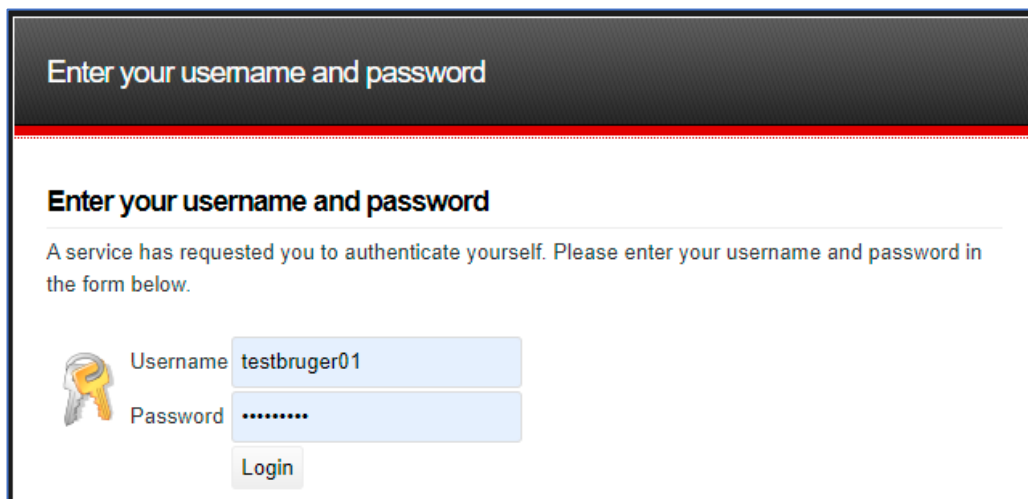
## 7.6. Test

Vi har tildelt en Jobfunktionsrolle til vores testbruger, der giver læseadgang til Klassifikation. For at teste går vi til [Klassifikation Brugerfladen](#), og bliver da bedt om at vælge Identity Provider.

Vi vælger vores egen:



Vi logger ind som testbruger01:



Og vi er dernæst logget ind i Klassifikation:

Du er logget ind som: Test Testesen, Ukendt CVR (11111111) Mine adgangsrettigheder Log ud

## Klassifikation

### Liste over klassifikationer

Eksportér Importér Konvertér Opret klassifikation

Brugervendt nøgle	Tilsluttede	Ejer (CVR)	Tilstand	Publiceret fra	Publiceret til
02.0003.010	Part IdTy	Ukendt CVR (11111111)	Publiceret		

Så snart din IdP er tilsluttet den fælleskommunale føderation, da kan du oprette og vedligeholde brugere og administrere adgange til alle brugervendte systemer i føderationen - på vegne af egen myndighed.

## 8. DATAAFGRÆNSNING PÅ ROLLER

### 8.1. Introduktion

Dataafgrænsning på en rolle angives som en "constraint":

```

<PrivilegeList ...>
  <PrivilegeGroup ...>
    <Privilege>{roleEntityId}</Privilege>
    <Constraint Name="{constraintEntityId}">...</Constraint>
  </PrivilegeGroup>
</PrivilegeList>

```

Hvis et brugervendt system (BVS) har angivet dataafgrænsning på dets brugersystemroller, da skal hver myndighed (kommune) vælge, hvorledes værdier for disse skal sættes, når deres brugere logger ind i systemet. Der er to muligheder:

- Statiske værdier
  - IdP sætter blot Jobfunktionsroller for bruger
  - Context Handler tilføjer dataafgrænsningen
- Dynamiske værdier
  - Sættes af Identity Provider (\*)

- Konverteres af Context Handler

(\*) Myndigheden vælger selv, hvorledes de ønsker at konfigurere deres IdP mht. hvorledes dataafgrænsning tildeles brugere - det er ikke et emne for denne vejledning.

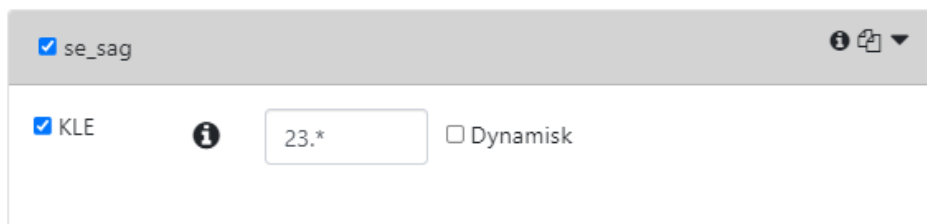
Når en myndighed opretter eller redigerer en Jobfunktionsrolle (JFR) i FK ADM, da får de mulighed for at tilføje Brugersystemroller (BSR) til pågældende JFR:

**+ Tilknyt brugersystemroller**

Det fremgår da, om pågældende BSR har dataafgrænsning. Og myndigheden kan vælge, om de ønsker at samme værdi skal sættes hver gang for pågældende JFR (statisk), eller om Context Handler skal aflæse værdien fra SAML-token bruger kommer med fra IdP (dynamisk). Hvilket er illustreret i følgende afsnit.

## 8.2. Statiske værdier

Her eksempel på en Brugersystemrolle med valgfri dataafgrænsning "KLE". Vi angiver en fast værdi:



The screenshot shows a configuration window for a user system role. At the top, the role name 'se\_sag' is displayed with a checkmark and a dropdown menu icon. Below this, there is a list of roles. The first role is 'KLE', which is checked. To its right is an information icon and a text input field containing the value '23.\*'. Further to the right is a checkbox labeled 'Dynamisk', which is currently unchecked.

Figur - Tilknytning af Brugersystemrolle med dataafgrænsning til Jobfunktionsrolle i FK ADM

Hvis du anvender din IdP i den fælleskommunale føderation, da skal du blot oprette samme JFR i dit brugerkatalog og dernæst tildele dine brugere. Alle brugere med denne JFR vil dernæst automatisk (af Context Handler) få Brugersystemrolle "se\_sag" med afgrænsning KLE = "23.\*" ved login i fagsystemet.

Hvis vi ønsker at emulere dette i en lokal føderation, da sørger vi for at vores dataafgrænsning er defineret i brugerkataloget:



### Dataafgrænsning

Liste Opret ny Dataafgrænsning #1

Navn

Entityld

Vi opretter brugersystemrollen og tilknytter dataafgrænsning:

### Roller

Liste Opret ny

Entityld

Dataafgrænsning  Følsomhed  KLE

Vi kan dernæst tilknytte denne rolle til en bruger og sætte en værdi for dataafgrænsning:

## IdP Brugere

Liste
Opret ny
Bruger #1

Navn	<input type="text" value="Test Testesen"/>
Serial	<input type="text" value="fd2ed2a9-09fc-4b4f-98a6-6d7bf206d088"/>
Brugernavn	<input type="text" value="testbruger01"/>
Adgangskode	<input type="text" value="Test1234!"/>
Rolle	<input type="checkbox"/> <a href="http://korsbaek.dk/roles/jobrole/KlassifikationLaesKorsbaek/1">http://korsbaek.dk/roles/jobrole/KlassifikationLaesKorsbaek/1</a>
Rolle	<input checked="" type="checkbox"/> <a href="http://mit-fagsystem.dk/roles/usersystemrole/se_sag/1">http://mit-fagsystem.dk/roles/usersystemrole/se_sag/1</a>
	<input type="text" value="23.*"/> <span style="float: right;">KLE</span>

Vi er således i stand til at emulere Context Handler, også når det gælder dataafgrænsning på brugersystemroller.

### 8.3. Dynamiske værdier

Hvis vi vælger "Dynamisk" dataafgrænsning på brugersystemrollen:

se\_sag i

KLE !

**EntityId** ?

http://korsbaek.dk/KLE/1/parametric

 Dynamisk

Da vil der automatisk blive defineret en dynamisk dataafgrænsningsparameter for vores myndighed, hvor vi skal angive navn. Her har vi valgt "KLE" og Entity Id bliver således:

http://korsbaek.dk/KLE/1/parametric

Vi har dermed fortalt Context Handler, at den skal aflæse værdien fra Constraint i brugers SAML-token Privileges\_Intermediate med dette entityld.

Vi opretter dataafgrænsning i vores brugerkatalog:

### Dataafgrænsning

Liste Opret ny

Navn

Entityld

JFR vi oprettede i FK ADM opretter vi også i vores brugerkatalog:

### Roller

Liste Opret ny

Entityld

Dataafgrænsning

- Følsomhed
- KLE
- KLE Parametric

Vi tildeler en bruger rollen:

Rolle

KLE Parametric

Når vi efterfølgende logger ind i fagsystem, der har pågældende Brugersystemrolle, da ser det ud som følger. Vores bruger kommer tilbage fra vores IdP med jobfunktionsrolle + dataafgrænsning:

## Privileges\_intermediate i token fra vores IdP

```
<PrivilegeList xmlns="http://itst.dk/oiosaml/basic_privilege_profile">
  <PrivilegeGroup Scope="urn:dk:gov:saml:cvrNumberIdentifier:11111111">
    <Privilege>http://korsbaek.dk/roles/jobrole/testrolle/1</Privilege>
    <Constraint Name="http://korsbaek.dk/KLE/1/parametric">12.13.14</Constraint>
  </PrivilegeGroup>
</PrivilegeList>
```

Context Handler vil dernæst oversætte denne JFR til "se\_sag" BSR, og samtidigt oversætte dataafgrænsningen til EntityID defineret af fagsystemet:

## Privileges\_intermediate i token fra Context Handler til Demo Brugervendt system

```
<bpp:PrivilegeList xmlns:bpp="http://itst.dk/oiosaml/basic_privilege_profile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <PrivilegeGroup Scope="urn:dk:gov:saml:cvrNumberIdentifier:11111111">
    <Privilege>http://demo-
brugervendtsystem.kombit.dk/roles/usersystemrole/se_sag/1</Privilege>
    <Constraint Name="http://sts.demo.dk/constraints/KLE/1">12.13.14</Constraint>
  </PrivilegeGroup>
</bpp:PrivilegeList>
```

Bemærk, at datagrænsninger også kan sættes som SAML-attributter. Men det vil kræve konfiguration i IdP-produktet samt udvidelse af brugerkataloget, så det er ikke understøttet i denne version. Du kan selvfølgelig selv tilføje det, hvis du har brug for det.

## VERSIONSHISTORIK

Version	Dato	Ændringer
1.0	30-06-2022	Første version