DIGITALISERINGS KATALOGET

KOM GODT I GANG

CERTIFIKATER

En trin for trin guide til dig, der skal bestille og konfigurere certifikater

September 2023

KOMBIT

Kommunernes it-fællesskab

1. Introduktion

I den fælleskommunale infrastruktur anvendes certifikater til Fælleskommunalt Adgangsstyring for systemer og til Fælleskommunalt Adgangsstyring for brugere. Certfikater anvendes til at sikre parters rette identitet, samt til at etablere sikker kommunikation mellem parterne. De anvendes, når et fagsystem integrerer med webservices og Fælleskommunal Beskedfordeler (BFO), samt når brugervendte systemer og Identity Providers integrerer med Context Handler. De er således helt centrale for sikkerhedsmodellen i infrastrukturen.

Formålet med denne guide er, at give dig en introduktion til de hyppigt forekommende emner ved ibrugtagning, så du kommer hurtigt og godt i gang. Guiden henvender sig primært til leverandører, der skal integrere med den fælleskommunale infrastruktur for første gang.

De første afsnit er generelle og beskriver forskellen på de offentlige/private versioner, den anvendte standard, samt hvordan du bestiller. De følgende afsnit beskriver, hvordan du registrerer og anvender et certifikat. Guiden indeholder følgende afsnit:

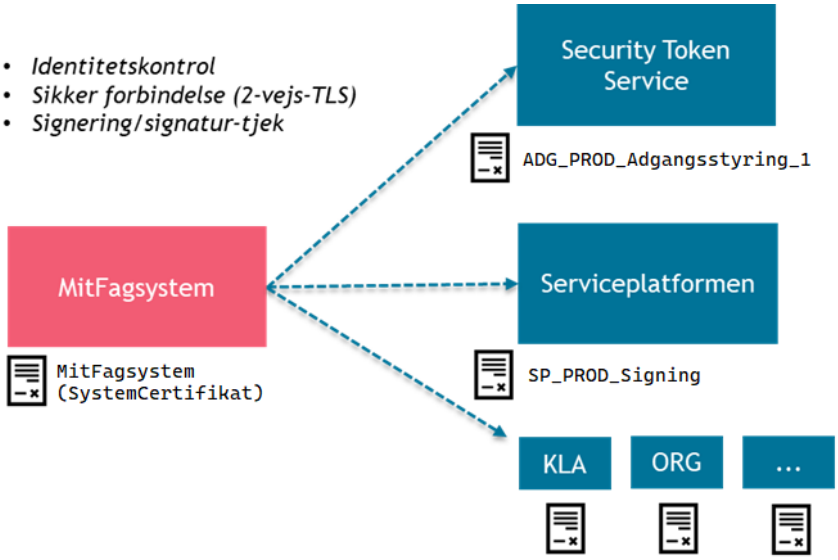
1. Introduktion
2. Offentlig vs. privat version
3. Systemcertifikater
4. Bestilling hos MitID Erhverv
5. Registrering i Fælleskommunalt Administrationsmodul (ADM)
6. Windows Certificate Store
7. Java Keystore
8. Certification Authorities
9. Konfiguration af 2-vejs TLS
10. Certificate Revocation Lists

- Du må ikke anvende samme certifikat på et IT-system til både testmiljø og produktionsmiljø - Du skal bestille og registrere dedikeret certifikat til hvert unikke IT-system og miljø.
- For OCES3-certifikater: Du skal bruge produktionscertifikater i både Eksternt testmiljø og Produktionsmiljø. Et anvendersystem, der skal teste mod Digitaliseringsstyrelsens DevTest4-miljø, kan dog bruge testcertifikat i Eksternt testmiljø.
- Du skal navngive dine certifikater i henhold til hvert IT-systems navn, så anvendelse og relation er til at gennemskue.

Et system består oftest af flere komponenter. Typisk en website backend, website frontend samt batch-jobs. Certifikatet, du registrerer på dit IT-system i ADM, dækker alle dets komponenter. Du kan således benytte samme certifikat til *Anvendersystem* og *Brugervendt system*. Men - du kan ikke benytte samme funktionscertifikat til *Brugervendt system* og *Identity Provider*.

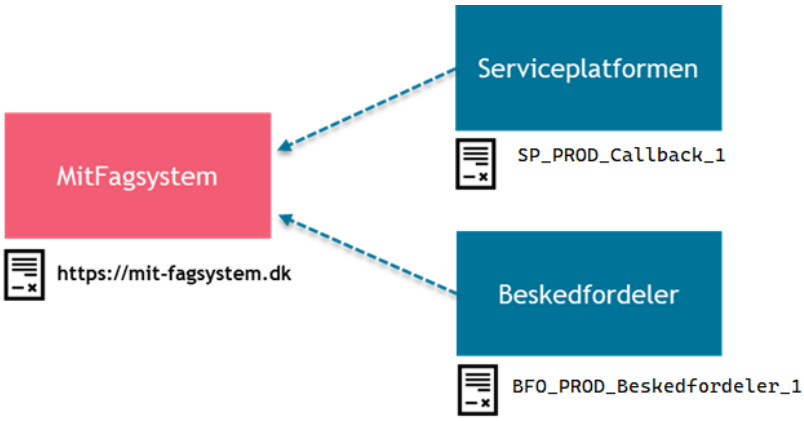
De individuelle komponenter i den fælleskommunale infrastruktur benytter hver eget certifikat. Security Token Service benytter certifikat "ADG_*<miljø>_Adgangsstyring_<rev>*". Services udstillet via serviceplatformen benytter certifikat "SP_*<miljø>_Signing_<rev>*". Certifikaterne er navngivet i henhold til deres anvendelse.

Når du kalder en service identificerer du dig med dit certifikat, og der etableres en sikker forbindelse med 2-vejs-TLS. For SOAP-services vil svaret yderligere være signeret med komponentens certifikat.



Hvis du udstiller en webservice mod fx Fordelingskomponenten, NgDP SF1606 eller en ØiR-integration, her vil Serviceplatformen kalde med dens callback-certifikat. Hvis du benytter PUSH-aflevering med Beskedfordel

er, her vil den kalde med eget certifikat. Bemærk, at der etableres 2-vejs-TLS til dit HTTPS-certifikat (*ikke* dit funktionscertifikat).



Du finder dem alle på [listen over anvendte certifikater]() og kan hente dem herfra. Se i øvrigt afsnit 9 Konfiguration af 2-vejs TLS.

For at dit rammeværk skal acceptere certifikaterne, skal du tjekke, at de respektive Certification Authorities (CA) er tilføjet dit Trust Store - se afsnit 8 Certification Authorities.

2. Offentlig vs. privat version

Grundlæggende findes certifikater i to versioner; med eller uden en privat nøgle. De anvendes til kryptering, således at afsender er sikker på, at det kun er tiltænkte modtager, der kan læse informationen. Certifikater anvendes ligeledes til signering, således at modtager kan være sikker på, at det var rette afsender, der sendte data. Signering sikrer også integritet; hvis data ændres under transporten vil signaturen ikke længere være valid.

Indehaver af versionen med den private nøgle kan således læse information, som kun er tiltænkt certifikatets ejer, og indehaver af versionen med den private nøgle kan udgive sig for at være certifikatets ejer og sende information på dennes vegne. Det er derfor kritisk, at du er bevidst om forskellen på de to, samt at du beskytter den private version på behørig vis. Derudover er det vigtigt, at du ikke ikke deler den private version med andre.

Da signering og kryptering foregår i begge retninger ved kommunikation mellem systemerne, skal hver part have registreret modpartens offentlige version af deres certifikater. Det er derfor, du skal registrere den offentlige version i Fælleskommunalt Administrationsmodul (ADM) for dit anvendersystem, brugervendte system eller Identity Provider. For de to sidstnævnte er certifikatet indlejret i SAML-metadata. Du skal registrere infrastrukturens offentlige version af dets certifikater på de systemer, som du kalder fra. Dem henter du i Digitaliseringskataloget.

Hvis du kommer til at dele den private version ved en fejltagelse, skal du straks tilbagekalde certifikatet (revocation) og få et nyt udstedt.

3. Systemcertifikater

Den offentlige standard for certifikater betegnes OCES-standarden, som er defineret af Digitaliseringsstyrelsen.

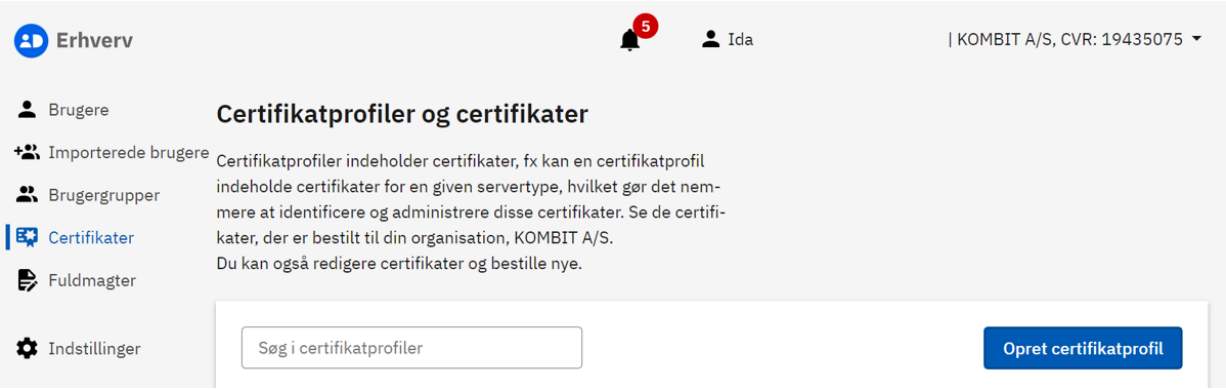
Infrastrukturen anvender OCES-certifikater, også kaldet organisationscertifikater. Det er også muligt at anvende systemcertifikater, som er en specialicering af organisationscertifikatet, til at repræsentere et specifikt system i din organisation.

4. Bestilling hos MitId Erhverv

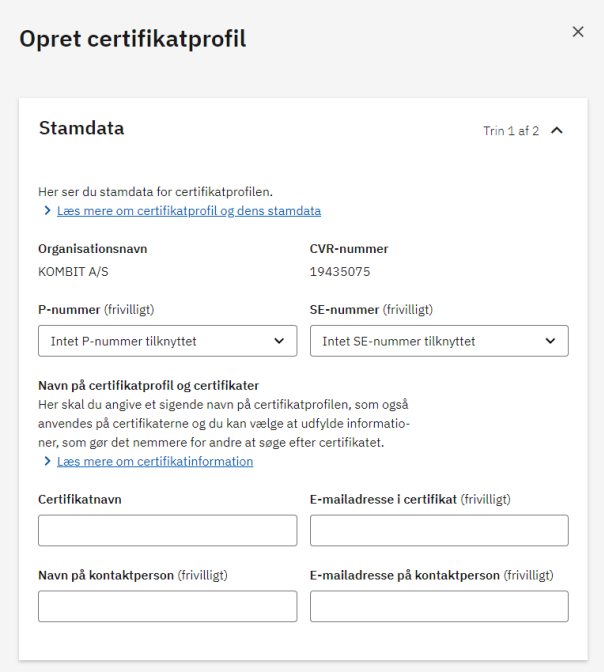
Infrastrukturen anvender produktionscertifikater i både Ekstern Test og produktions-miljøet. Bestilling af OCES-certifikater foregår via MitID Erhvervs hjemmeside og kan kun foretages af en MitID Erhverv-administrator fra virksomheden.

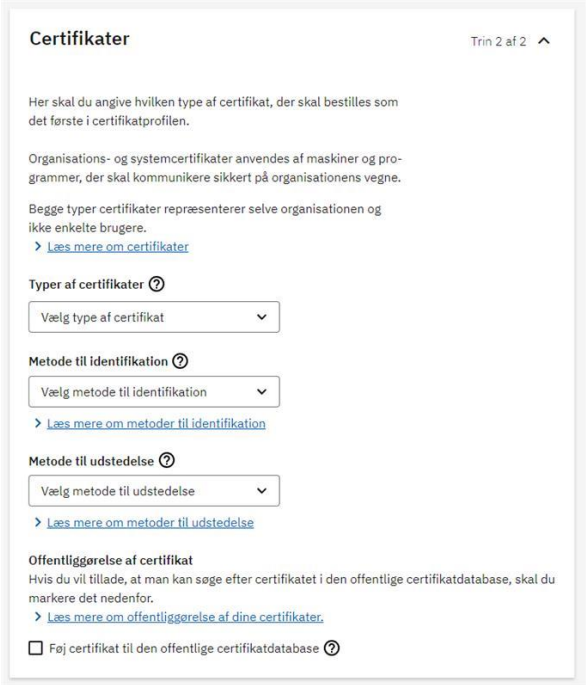
Bestilling af certifikat på MitIDs portal

I MitID Erhvervs selvbetjeningen vælger du menupunktet ”Certifikater”. Her kan du bestille og administerer certifikater for din organisation.



I MitID Erhvervs administrations portal kan du bestille certifikater ved at trykke på ”opret certifikatprofil”.





Udfyld detaljer omkring certifikatet og åbn derefter trin 2.

Vælg certifikat typen og udstedelses metoden. Ved valg af engangskode som udstedelsesmetode, husk da at gemme koden.

Du får en adgangskode under udstedelse af certifikatet som er nødvendig for at kunne anvende certifikatet. Husk at notere adgangskoden og gem den sikkert.

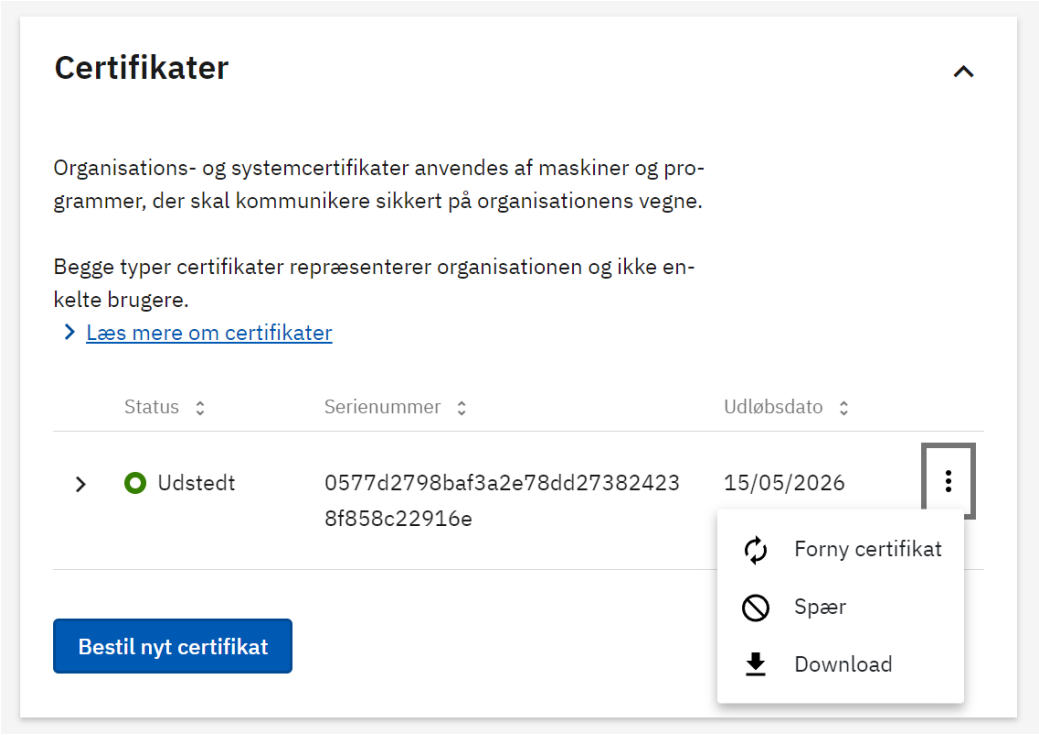
Når du aktiverer og udsteder et certifikat, får du udleveret versionen med den private nøgle som er gemt i PKCS#12 (.p12/.pfx), der er et generelt format, der fungerer på tværs af platforme.

Bemærk, at du efterfølgende kun kan hente den offentlige version uden den private nøgle.

Hvis du mister versionen med den private nøgle, eller mister adgangskoden til den, da kan du ikke længere benytte certifikatet, og du skal anmode om at få et nyt certifikat udstedt. Du kan ikke genetablere adgangskoden.

Administration af certifikater

På MitID Erhvers selvbetjening kan du fremsøge og vælge dit certifikat:



Ved at trykke på de tre prikker ud for certifikatet, eller blot trykke på rækken, kan du administerer det enkelte certifikat:

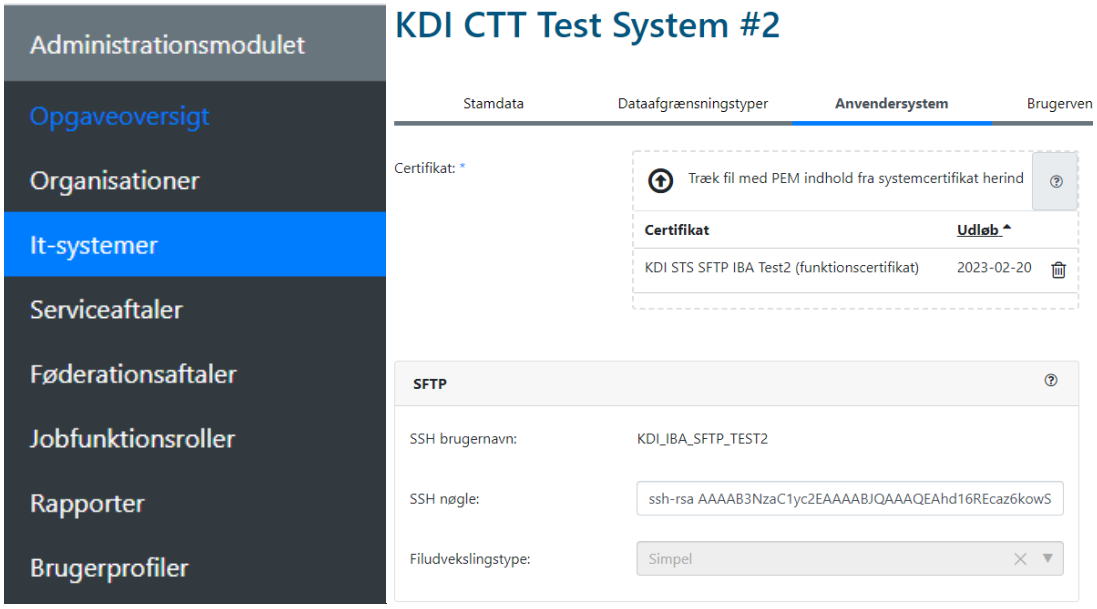
- Her kan du Forny Certifikatet, hvilket genererer en ny version med ny udløbsdato (Fornys et certifikat, spæres det gamle ikke automatisk).
- Ved at trykke på Download henter du den offentlige version.
- Du kan spærre certifikatet.

Certifikaterne udløber automatisk ved udløbsdatoen og kan ikke benyttes herefter. Når man genudsteder et certifikat, så udstedes der en ny ”version” af certifikatet, så begge certifikater er aktive på samme tid. Dette giver dig mulighed for i god tid at skifte certifikatet, inden det gamle udløber. Det gamle certifikat kommer således ikke automatisk på revocation-list, med mindre man specifikt anmoder om dette, i stedet udløber det blot.

Den offentlige version hentes som en CER fil (.cer) i PEM-format. Det er denne, du skal registrere på dit anvendersystem i ADM. I tilfælde af brugervendt system eller Identity Provider, da vil du efter lokal konfiguration med det private certifikat kunne udtrække SAML-metadata som har den offentlige version indlejret. Det er således SAML-metadata-filen du registrerer i ADM.

5. Registrering i Fælleskommunalt Administrationsmodul (ADM)

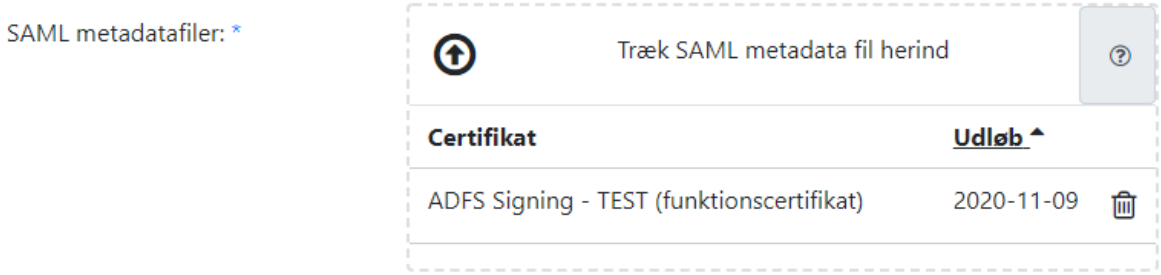
Når du har modtaget dit certifikat, skal den offentlige version registreres i ADM (test eller produktion). Hvis du skal integrere med webservices eller Beskedfordeler, skal certifikatet registreres på Anvendersystem. Her trækker du blot certifikatet (.cer/.pem) ind i boksen med den stiplede linje og klikker på "Gem" knappen.



PEM-formatet er tekst der starter med "----- BEGIN CERTIFICATE -----". CER filer findes både i binært og PEM-format, så du kan ved at kigge i filen se hvilket format, det har. Den version du henter fra Nets er allerede i det rigtige PEM-format.

Det nemmeste er at hente den offentlige version fra Nets. Du kan også anvende OpenSSL til at generere den offentlige version, eller anvende Windows Certificate Snap-in.

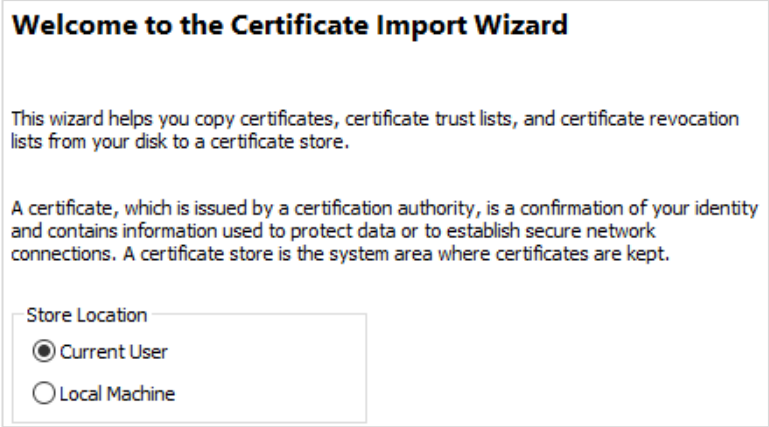
For brugervendte systemer og Identity Providers bliver certifikatet automatisk registreret, når du uploader SAML-metadata filen, da det er indlejret i denne.



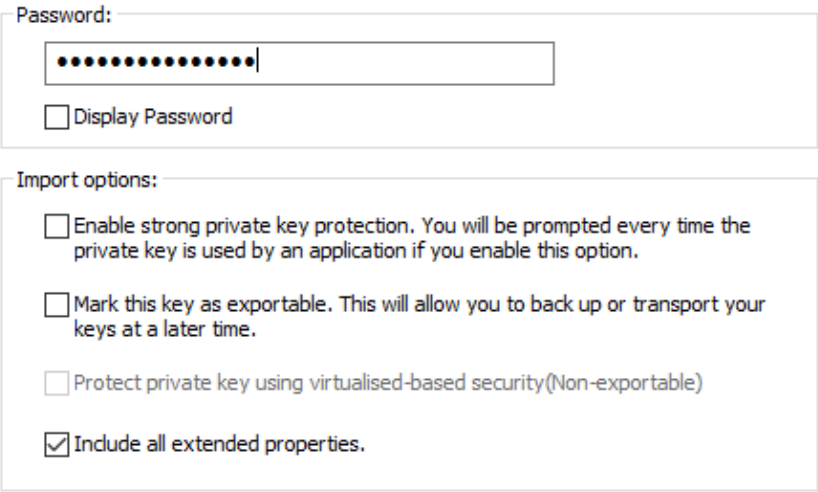
Opdateringer bliver automatisk provisioneret til Security Token Service (Adgangsstyring for systemer) eller Context Handler (Adgangsstyring for brugere). Dette sker næsten umiddelbart, og du vil inden for kort tid kunne bruge dit certifikat.

6. Windows Certificate Store

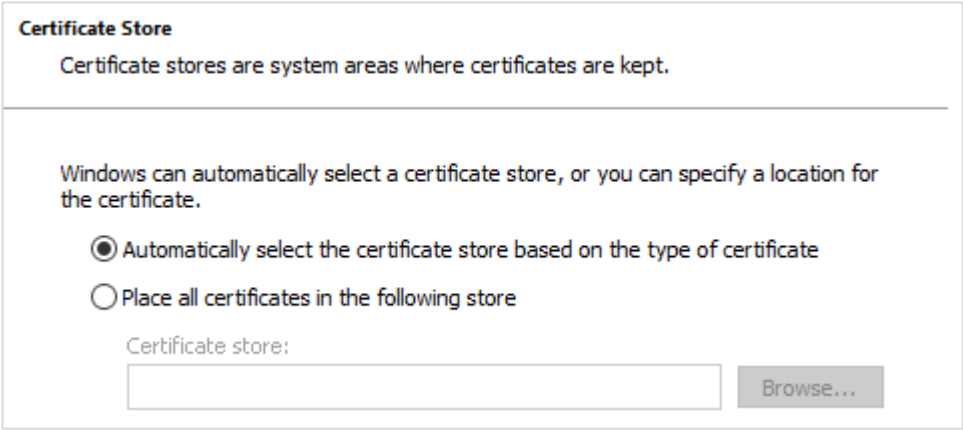
Håndtering af certifikater på Java-platformen er beskrevet i efterfølgende afsnit. Du importerer et certifikat lokalt på en Windows-maskine ved at aktivere filen (dobbelt-klik eller <enter>). Dermed aktiveres Certificate Import Wizard:



Hvis du skal teste lokalt, så kan du anvende ”Current User”. Når koden skal afvikles fra en Windows-server, placerer du certifikater under ”Local Machine”, dermed er de tilgængelige for alle tekniske brugere, som programmer afvikles i context af. Du bliver derefter bedt om at indtaste koden til den private nøgle:

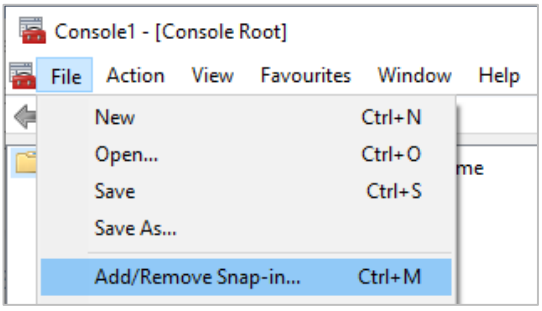


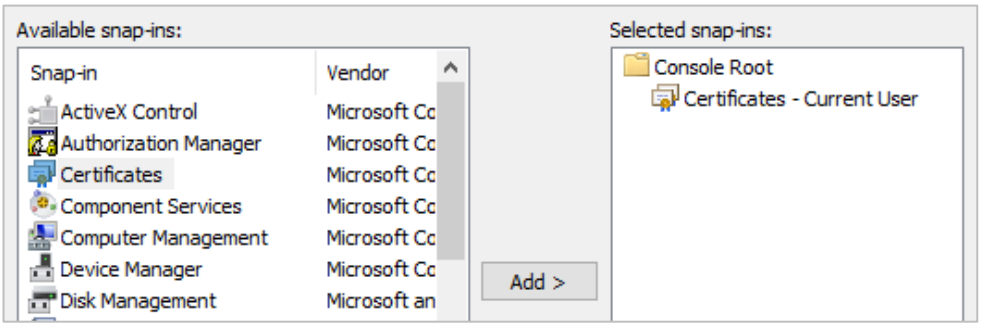
Som det næste bliver du spurgt om, hvor certifikatet skal placeres:



Vær opmærksom på, om jeres virksomhed har en standard vedrørende registrering og placering af certifikater, som skal følges. Og vær opmærksom på om den context (bruger), som processen kører i, har adgang til det certificate store der anvendes samt adgang til den private nøgle hvis eget certifikat.

For at tilgå Certificate Store startes Microsoft Management Console (mmc.exe). Dernæst tilføjes "Certificate" Snap-in:

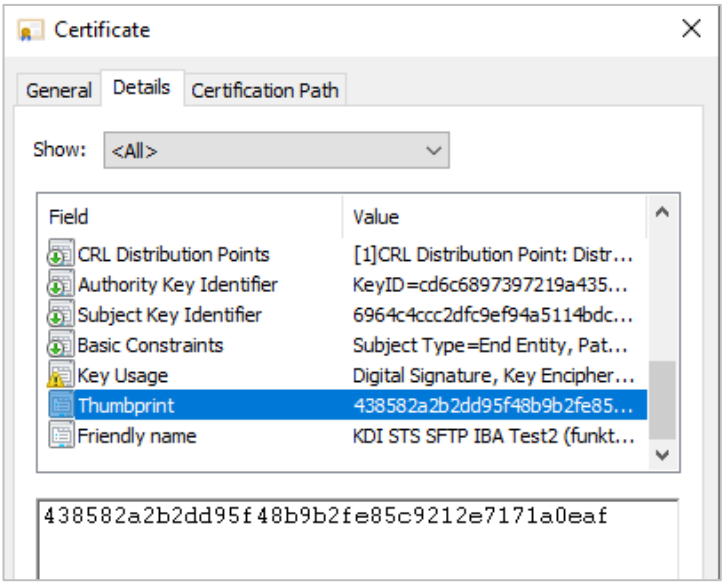




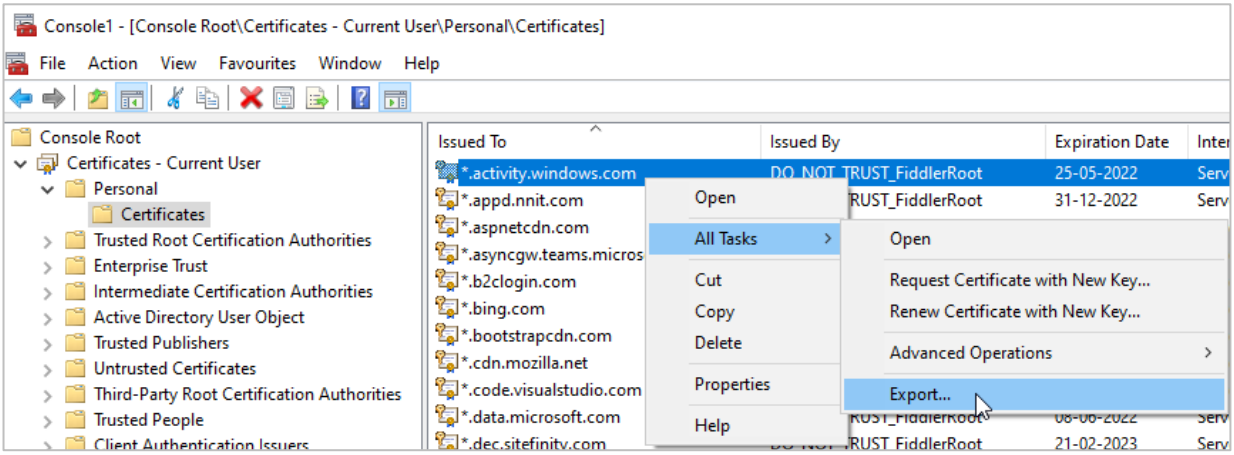
Herfra kan du:

- Eksportere et certifikat til offentlig version i PEM-formatet (.cer).
- Se detaljer for et certifikat.
- Se Certification Authority (CA) path/chain samt tjekke, at denne er valid.
- Se Thumbprint, som skal bruges i .NET kode.

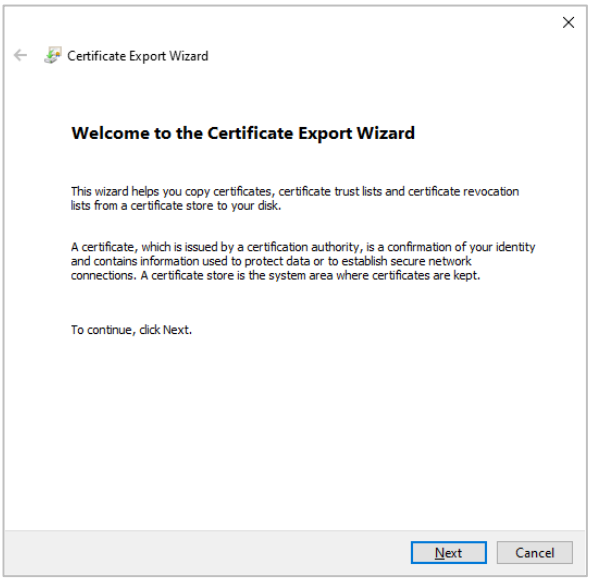
Når du dobbelt-klikker på et certifikat og vælger detaljer, finder du Thumbprint på listen af attributter. Thumbprint anvendes ofte, når koden laver opslag i Certificate Store for at hente et certifikat (der kan laves opslag på andre attributter også).

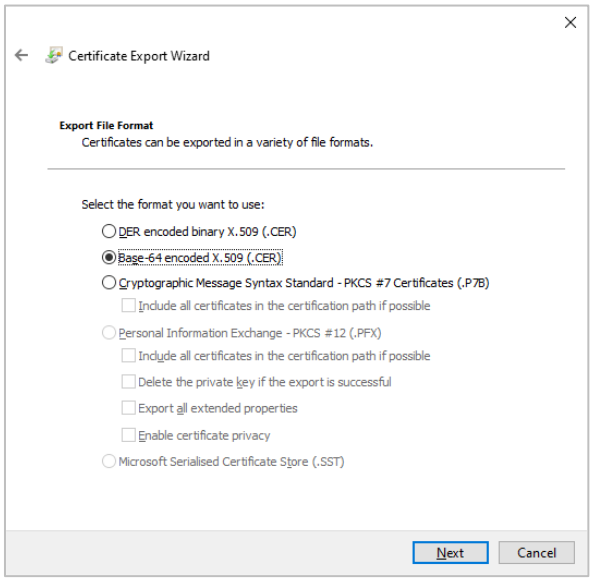


Højre-klik på et certifikat for at se muligheder:



Når du har importeret den *private* version (.p12/.pfx med privat nøgle), da kan du efterfølgende eksportere den *offentlige* version ved at vælge "Export..." i menu. Det starter en Wizard der ser ud som følger:





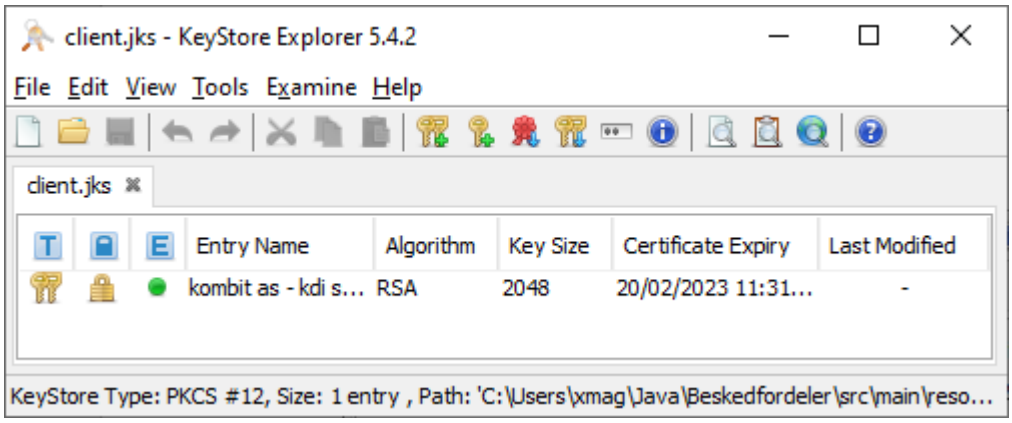
Husk at vælge "Base-64 encoded X.509" som format. Navngiv og gem filen hvor det passer. Det er denne version du skal uploade på dit anvendersystem i ADM.

Som beskrevet tidligere skal du skal hente og registrere infrastrukturens certifikater på den maskine koden afvikles fra, på samme måde som dit eget certifikat. Der findes separate certifikater til Security Token Service (Adgangsstyring for systemer), Context Handler (Adgangsstyring for brugere), webservices og de fælleskommunale støttesystemer. Du behøver selvfølgelig kun at registrere de certifikater, der tilhører komponenter, du skal integrere med.

Eksempler på anvendelse af certifikater ved kald til webservices findes i .NET client for Serviceplatformens DemoService.

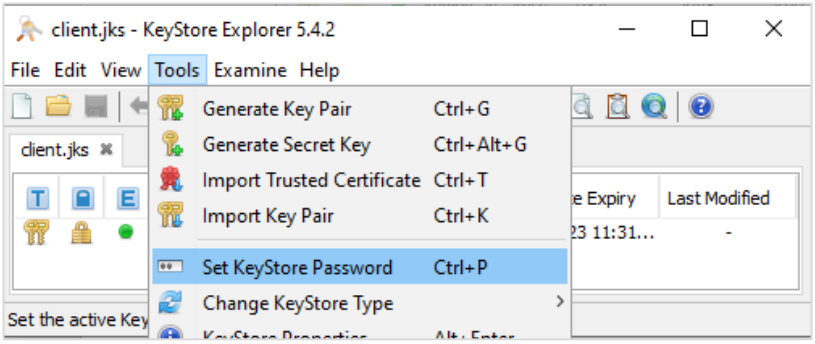
7. Java Keystore

Til håndtering af certifikater på Java-platformen kan du benytte https://keystore-explorer.org/. Du kan også anvende Keytool, der følger med Java SDK eller JRE. Certifikater gemmes i Java KeyStore filer (.jks).



Eksempler på anvendelse af certifikater ved kald til infrastrukturen findes i demo-kode til Beskedfordeler (i dokumentationspakken) samt Java client for Serviceplatformens DemoService. De anvender begge .jks filer, der kan genanvendes, blot man udskifter relevante certifikater, men du kan med fordel lave to nye tomme keystores og kalde dem fx client.jks og trust.jks. Husk at skifte referencer til dem i koden.

Filen *client.jks* indeholder det private certifikat, der bruges ved kald til infrastrukturen. Dette skal du erstatte med dit eget. Vigtigt: Et keystore kan indeholde flere certifikater, men client.jks må kun indeholde et certifikat for dit anvendersystem. Det skal have samme adgangskode som selve certifikatet. Vælg ”Tools” i menu og dernæst ”Set KeyStore Password”:



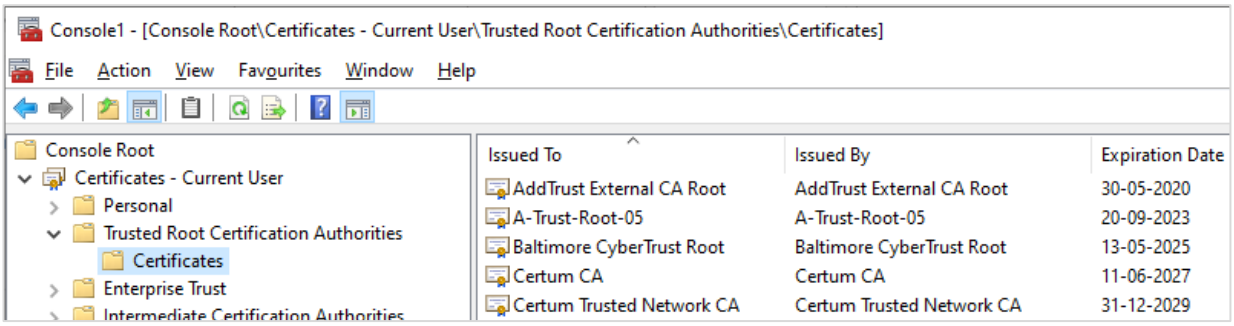
NB: Du skal vælge "*Import Key Pair*" når du skal tilføje den private version (p12/pfx) af dit funktionscertifikat til client.jks.

Filen *trust.jks* indeholder infrastrukturens certifikater for det eksterne testmiljø, og disse bør være gyldige, hvis du har hentet seneste version af demo-koden. Hvis de er udløbet, kan du hente seneste version fra infrastrukturens certifikater. Dette Keystore har ikke behov for at få sat en adgangskode, da det kun indeholder offentlige certifikater. Det er blot manglende CA du skal importere her, da certifikater anvendt til signing og krypteret forbindelse medsendes i svar fra endpoint.

8. Certification Authorities

For at et certifikat skal fungere korrekt, skal certification authorities (CA), der siger god for det, også være registreret. Der er oftest kun to niveauer i Certification Chain; *Root* og *Intermediate* CA. Det plejer kun at være nødvendigt, at registrere root CA, men der kan være situationer, hvor det også er nødvendigt at registrere og etablere tillid til Intermediate CA.

Hvis kode afvikles i .NET skal CA være registreret i *Trusted Root Certification Authorities* (Current User eller Local Computer afhængigt af den context programmet kører i)*:*



Hvis kode afvikles i Java kan CA registreres i \lib\security\cacerts som er en underfolder til dit Java Runtime Environment. Du kan se indholdet og vedligeholde certifikater i dette med *keytool*:

```
..\jdk1.8.0_261\jre\lib\security>keytool -list -keystore cacerts
```

Fordelen er, at det gælder alle java-applikationer der afvikles på maskinen. Alternativt kan du tilføje manglende CA til et Trust Keystore som du henviser til fra koden, som i eksempelkoden til webservices og Beskedfordeler.

Hvis dit Trust Store er af nyere dato, da indeholder det sandsynligvis allerede de nødvendige Root og Intermediate certifikater som indgår i Certification Chain for de certifikater der anvendes. Så det er ikke altid nødvendigt at tilføje CA til Trust Store. Det kan være nødvendigt hvis der anvendes self-signed certifikater, eller hvis Trust Store ikke er af nyere dato på den maskine koden afvikles fra.

Hvis et CA i certification path ikke er registreret på maskinen (Windows) vil det være markeret med et udråbstegn udfor pågældende certifikat under Certification Path. Følgende er et konstrueret eksempel der viser, hvordan du identificerer problemet og fikser det.

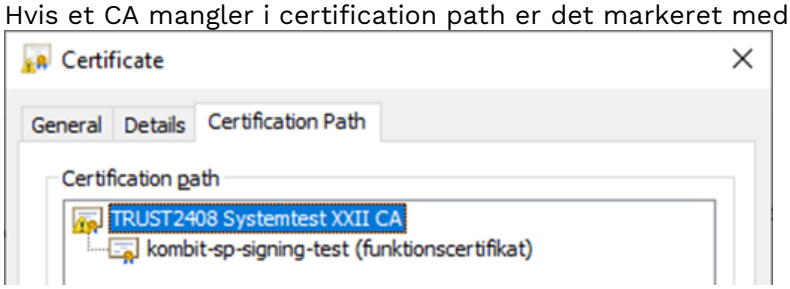
Følgende certifikat er *Chained*, det indeholder også Root og Intermediate CA. Du ser tre certifikater, hvis du åbner filen i en teksteditor:


```
-----BEGIN CERTIFICATE-----
MIIGITCCBQmgAwIBAgIEW6uwOTANBgkqhkiG9w0BAQsFADBIMQswCQYDVQQGEwJE
...
p+qspCBVu9Vru0UaGMNXyK0Ks5JIvHbYmL0RC2VASWItvePcft+mQKc6iBO+HCnp
WCtjVk5FlcfLxyV96UIUkqW4QABV1/2E0K0OsRppRTe51Z0EWp/Meid7ldMCSKtR
5e6+hgihN4nj8QQgk/jJJ9CDKDommMw9+rhG9VRh5kEeORmQ6A==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFQTCCAymgAwIBAgIEWBh+dDANBgkqhkiG9w0BAQsFADBPMQswCQYDVQQGEwJE
...
qFZ4ofDOwsr9fQ2jN+vdIX2ZPUlK5KBZ9Lo2CikaEQsxXLOv6PfBZqo6ukkOeqTq
nVYCW68=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGSDCCBDCgAwIBAgIES+pulDANBgkqhkiG9w0BAQsFADBPMQswCQYDVQQGEwJE
...
d/kID32R/hJPE41o9+3nd8aHZhBy2lF0jKAmr5a6Lbhg2O7zjGq7mQ3MceNeebuW
XD44AxIinryzhqnEWI+BxdlFaia3U7o2+HYdHw==
-----END CERTIFICATE-----
```

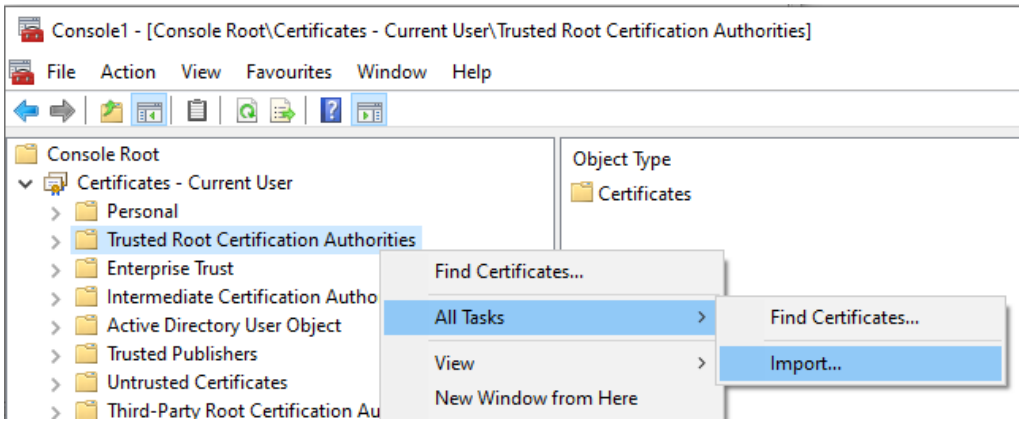

Hvis du gemmer de tre certifikater i hver sin .cer fil, aktiverer dem én efter én for at se detaljer, kopierer navnet fra ”Subject, Common Name (CN)” og omdøber filen efterfølgende, da får du følgende tre certifikater som du kan importere individuelt (fra eksemplet):

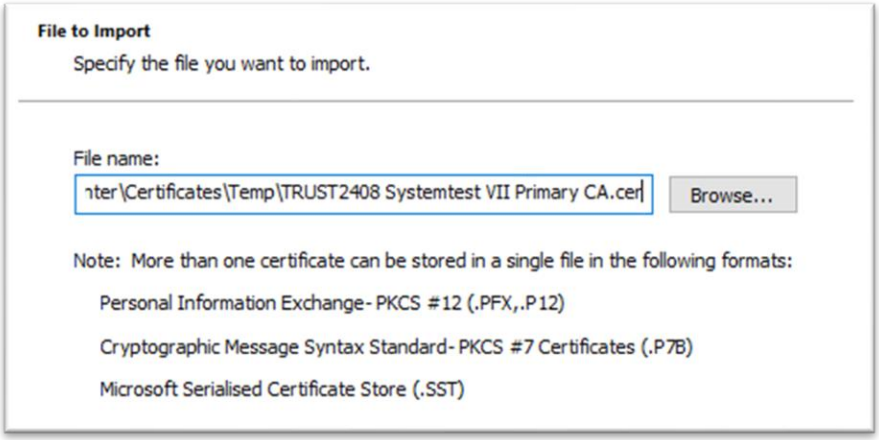
- Eksempel certifikat_KDI1.cer
- Den Danske Stat OCES rod-CA.cer
- Den Danske Stat OCES Udstendende-CA 1.cer

Hvis et CA mangler i certification path er det markeret med et gult udråbstegn.

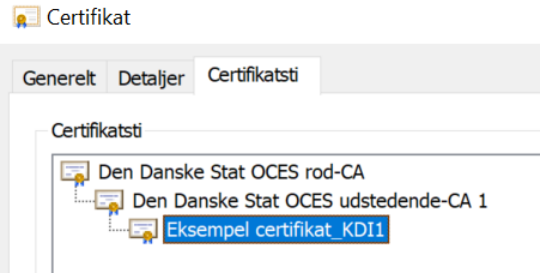


Vi skal da importere det manglende CA under *Trusted Root Certification Authorities*:

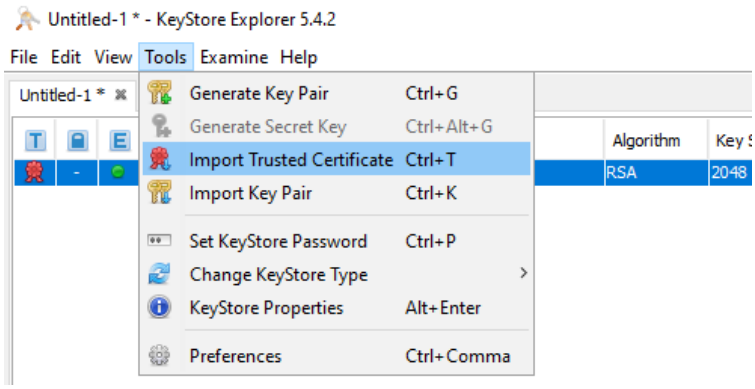




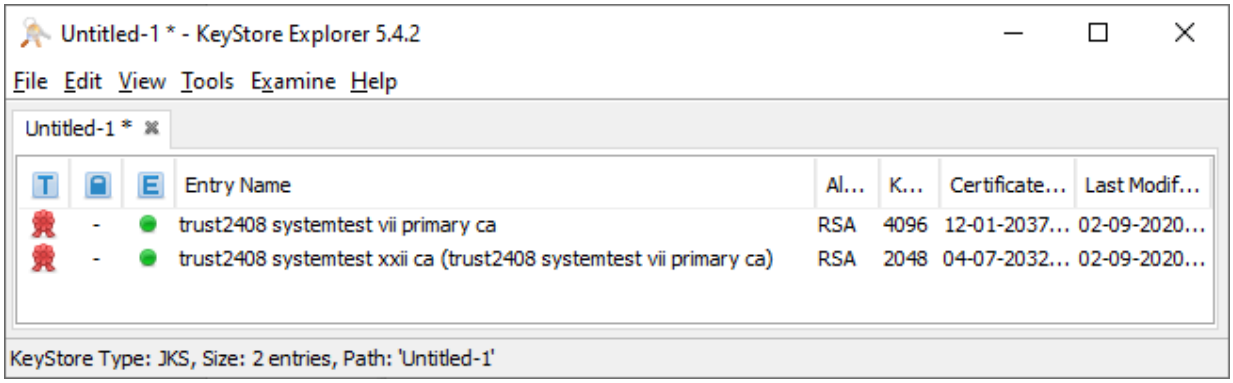
Det manglende CA er nu registreret og vi kan se at certification path er validt:



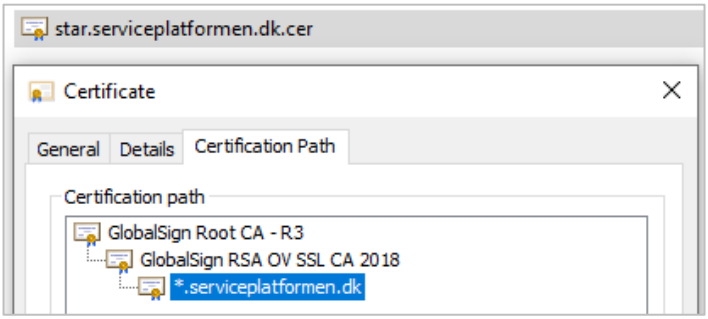
I et Java-miljø skal manglende CA importeres i det generelle eller lokale Trust store. Følgende eksempel viser proceduren for tilføjelse af CA til det lokale Trust store. Vælg import fra Tools menu og importér de to CA certifikater du gemte i separate filer.



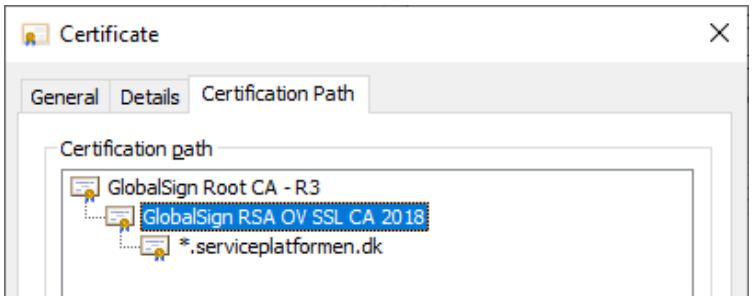
Her eksempel hvor vi har importeret de to CA tilhørende certification path for certifikat der benyttes til signering af beskeder i exttest:



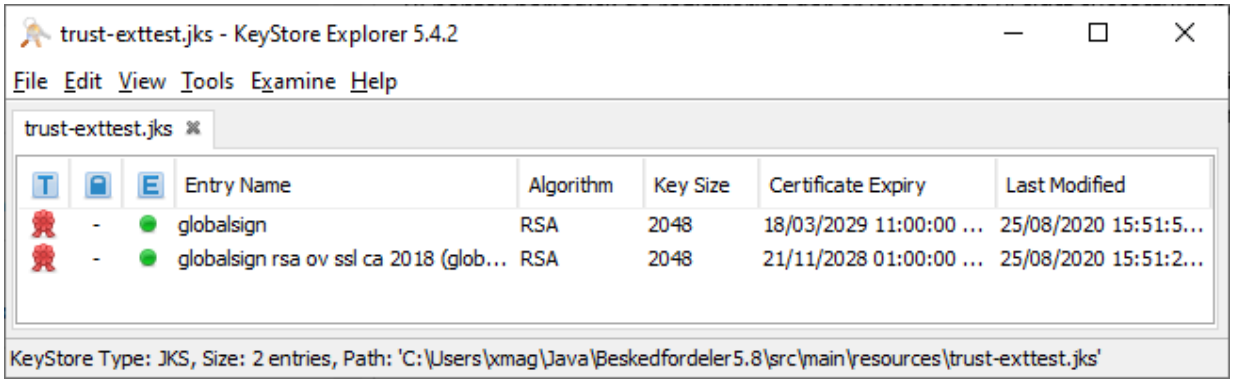
Ikke alle certifikater er *chained*, dvs. også indeholder de tilhørende CA. I følgende tilfælde findes de to CA allerede er på listen af Trusted root certificates i Windows.



Hvis disse CA mangler i dit Java Trust store, da kan du eksportere de to CA og efterfølgende importere dem i dit Java keystore. Du kan åbne CA fra certification path og derfra eksportere det, som illustreret tidligere i dokumentet.



Her er de to CA tilhørende HTTPS-certifikatet for serviceplatformen eksporteret fra Windows Certificate Store og importeret i lokalt Java keystore:



9. Konfiguration af 2-vejs TLS

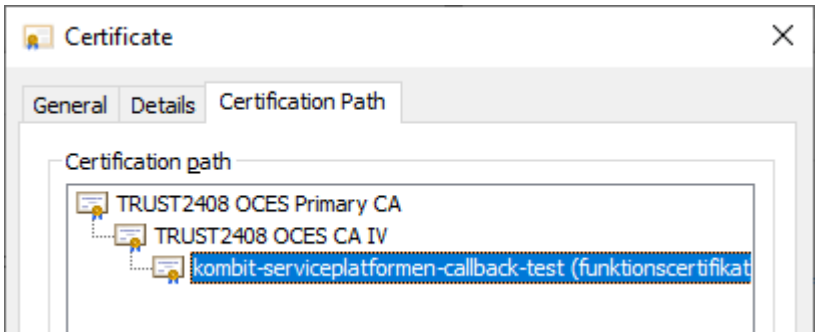
Dette er påkrævet, når du udstiller en webservice mod den fælleskommunale infrastruktur, i forbindelse med at du skal:

- Modtage PUSH-beskeder fra Beskedfordeler
- Modtage kvitteringer/fordelingsobjekter fra Fordelingskomponenten
- Modtage forespørgsler fra ØiR-integrationerne
- Modtage digital post via SF1606

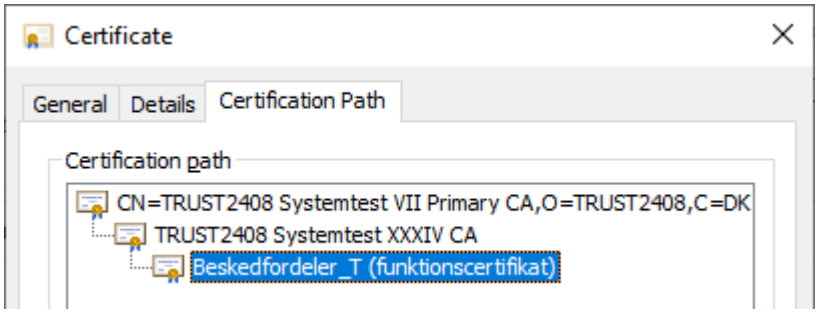
Dette er også kendt som klient-certifikat autentificering. Ved 2-vejs TLS benytter begge parter i kommunikationen, klient og service, et certifikat og der etableres gensidig tillid. Med andre ord, Serviceplatformen eller Beskedfordeler identificerer sig med et certifikat, når den kalder din webservice.

Bemærk, at når du som klient kalder en service i den fælleskommunale infrastruktur, da benytter du et funktionscertifikat, mens at service benytter et HTTPS-certifikat (også kaldet TLS eller SSL certifikat). Her er det omvendt; du udstiller dit endpoint med et HTTPS-certifikat og dit endpoint kaldes med et funktionscertifikat.

Fordelingskomponenten, ØiR integrationer og Digital Post identificerer sig med serviceplatformens callback-certifikat:



Beskedfordeler identificerer sig med eget certifikat:



Yderligere information om netværkssikkerhed

I Digitaliseringskataloget finder du information om anvendte ciphers og IP-adresser der kaldes fra. Hvis du ikke modtager kald til dit endpoint, check da:

- At ældre protokoller såsom TLS 1.1, TLS 1.0 og alle versioner af SSL, er deaktiveret
- At dit HTTPS-certifikat er validt, dvs. matcher hostnavn og ikke er udløbet
- Hvis dit endpoint udstilles gennem en firewall/load-balancer, at
    - Denne er konfigureret til 2-vejs-TLS
    - Eller gennemstiller HTTPS-negotiation direkte
- At du selv kan etablere en sikker forbindelse til dit endpoint, når du kalder med dit eget funktionscertifikat.
- At test fra SSLLabs ikke viser fejl eller kritiske advarsler
- At Root Certification Authorities er tilføjet trust store

Praktisk eksempel

Følgende eksempel er udført med Nginx og PHP, for at illustrere de trin du skal igennem. De praktiske opgaver afhænger af din valgte platform og teknologier, men det er samme opgaver der skal udføres.

Vi kan se, at root CA er "*TRUST2408 OCES Primary CA*" når der kaldes fra Fordelingskomponenten, og vi skal etablere tillid til dette for at callback-certifikatet accepteres. Vi eksporterer dette i PEM-format og gemmer i en tekst-fil "trusted_ca.pem". Dernæst opdaterer vi webserver-konfigurationen:

1. Vi aktiverer klient-certifikat autentificering
2. Vi sætter tilladte protokoller til TLS version 1.2 og 1.3
3. Vi henviser til vores Certification Authority trust-store "trusted_ca.pem"

```
server {
        ...
        ssl_verify_client on;
        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_client_certificate /etc/nginx/client-certs/trusted_ca.pem;
        ...
}
```

Vi tilføjer følgende konfiguration, for at kunne aflæse detaljer om certifikat der kaldes med:

```
server {
        ...
        location ~ \.php$ {
                ...
                fastcgi_param SSL_CLIENT_VERIFY $ssl_client_verify;
                fastcgi_param SSL_CLIENT_S_DN $ssl_client_s_dn;
                fastcgi_param SSL_CLIENT_I_DN $ssl_client_i_dn;
```

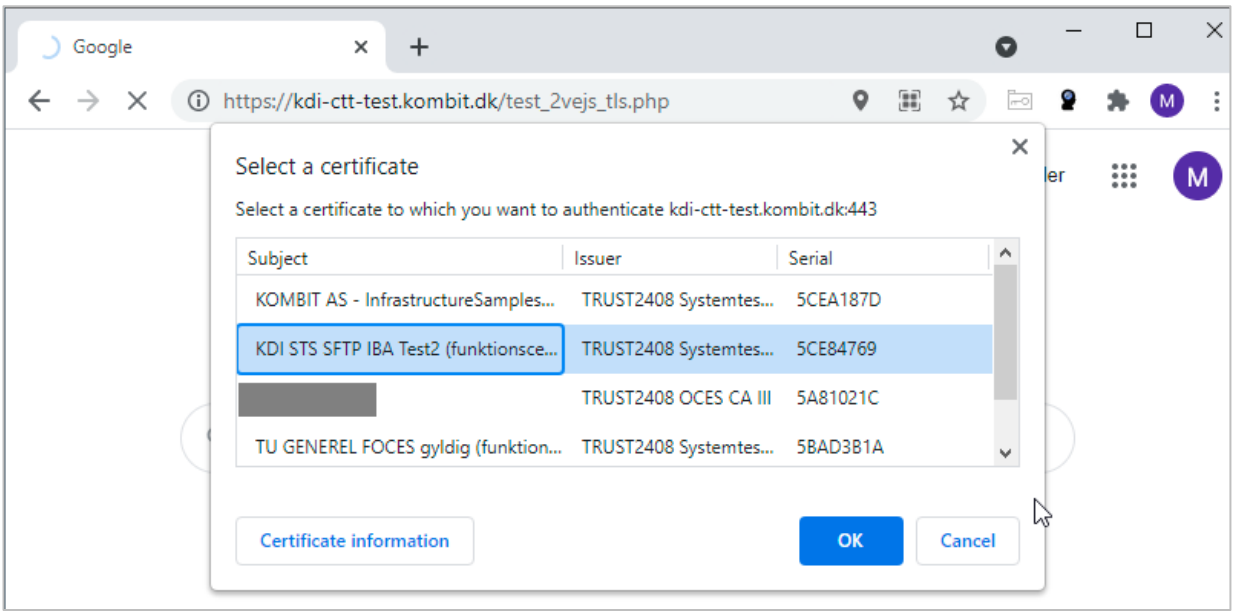
```
                fastcgi_param SSL_PROTOCOL $ssl_protocol;
                fastcgi_param SSL_CLIENT_SERIAL $ssl_client_serial;
                fastcgi_param SSL_CLIENT_V_END $ssl_client_v_end;
                fastcgi_param SSL_CLIENT_V_REMAIN $ssl_client_v_remain;
                fastcgi_param SSL_CLIENT_FINGERPRINT $ssl_client_fingerprint;
        }
}
```

Vi laver et PHP-script, der udskriver information om certifikat klienten kalder med:

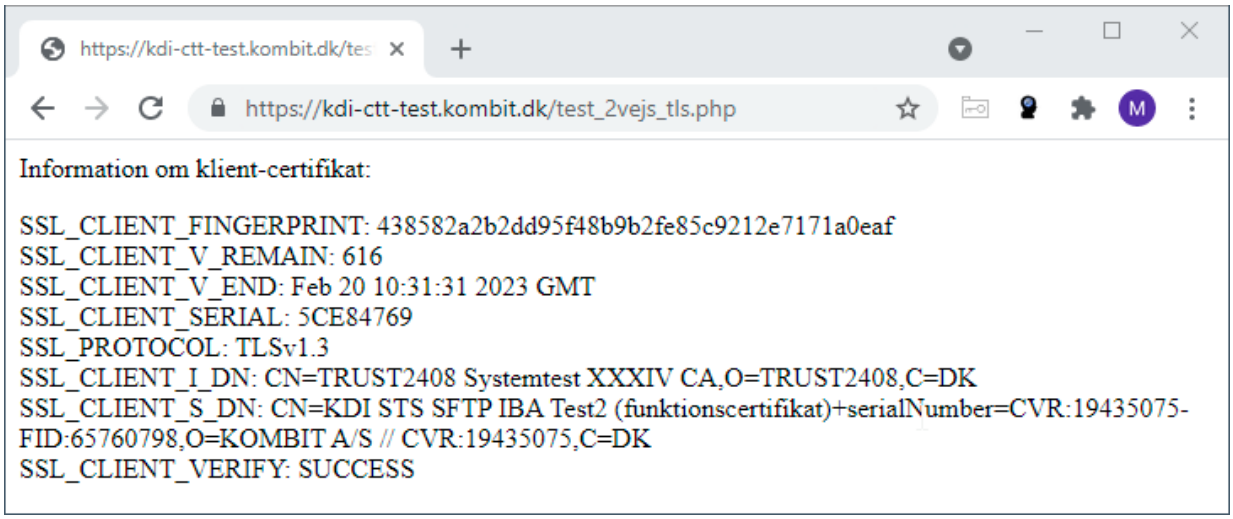
test_2vejs_tls.php

```
<html>
<body>
<p>Information om klient-certifikat:</p>
<?php
foreach ($_SERVER as $name => $value) {
   if (preg_match('/^SSL_/',$name)) {
      echo("$name: $value</br>");
   }
}
?>
</body>
</html>
```

Vi kalder script fra vores browser. Browser identificerer, at klient-certifikat er påkrævet, og beder os vælge certifikat, der skal kaldes med. Som en del af HTTPS-negotiation starter webserver med at præsentere de CA den stoler på, og din browser vil kun foreslå klient-certifikater på din maskine, som er med på denne liste. Vi vælger et certifikat:

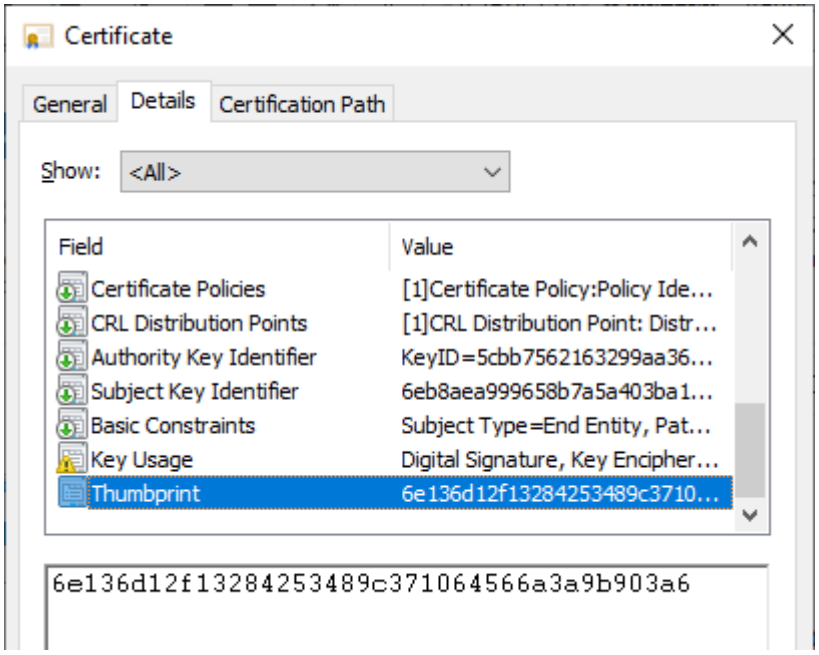


Her er resultatet:



Vi er nu i stand til at aflæse Thumbprint/Fingerprint fra certifikat, der anvendes i forespørgsel, og anvende dette til at sikre, at det er Serviceplatformens callback-certifikatet der kalder.

Vi aflæser callback-certifikatets fingerprint fra detaljer:



Vi kan nu tilføje kode i vores webservice der verificerer, at det er Serviceplatformen der kalder:

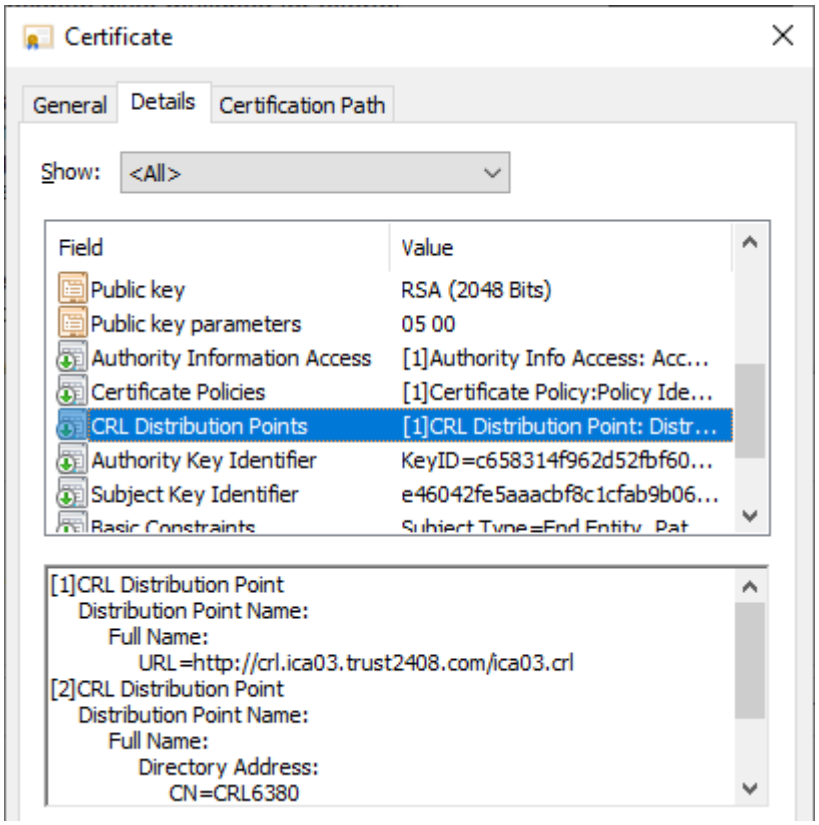
```
define('SP_CERTIFICATE_FINGERPRINT, '6e136d12f13284253489c371064566a3a9b903a6');

if ($_SERVER['SSL_CLIENT_FINGERPRINT'] != SP_CERTIFICATE_FINGERPRINT) {
    // Logning, afvisning, notifikation
}
```

Dette er blot et enkelt eksempel. Du bør selvfølgelig definere værdien som en konfigurationsparameter. Samt tillade flere værdier, hvis du fx samtidigt skal tillade et nyt kommende certifikat, når det nuværende er ved at udløbe, eller skal teste med at kalde med eget funktionscertifikat.

10. Certificate Revocation Lists

En ofte benyttet praksis for servere i en DMZ er, at de som udgangspunkt og af sikkerhedsmæssige årsager ikke har lov til at kalde ud. Så hvis du får en fejlbesked, at et certifikat ikke kan valideres, tjek da også, at koden har adgang til den relevante Certificate Revocation List (CRL). Under *Details* kan du se *CRL Distribution Points*:



Vi ser her, at CRL for serviceplatformens signing-certifikat findes på adressen http://crl.ica03.trust2408.com/ica03.crl. Tjek da, at der tillades trafik fra server til dette endpoint.

Af ren nysgerrighed kan vi hente CRL og se indholdet med OpenSSL:

```
C:\OpenSSL\x64\bin>openssl.exe crl -in C:\Users\xmag\Downloads\ica03.crl -inform DER -text -
noout
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = DK, O = TRUST2408, CN = TRUST2408 OCES CA III
        Last Update: Jun 28 07:18:52 2022 GMT
        Next Update: Jun 28 19:20:52 2022 GMT
```

```
        CRL extensions:
            X509v3 CRL Number:
                1156375
            X509v3 Authority Key Identifier:
                keyid:C6:58:31:4F:96:2D:52:FB:F6:0B:78:F7:CA:DC:1E:D8:DA:BC:A3:84

Revoked Certificates:
    Serial Number: 5A95F398
        Revocation Date: Jun 21 11:59:37 2022 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
                Unspecified
...
```