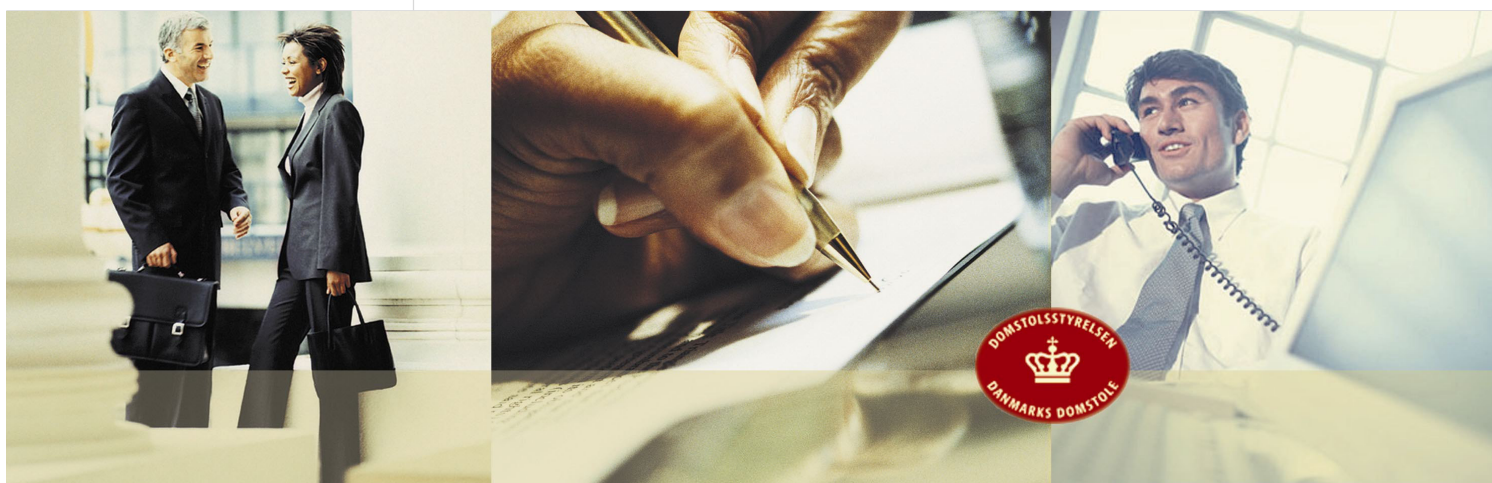


e-Tinglysningsprojektet

Kommunikationstest



Indholdsfortegnelse

1 Kommunikationstest	2
1.1 KommunikaionTest	5
1.2 KommunikationTestWSS	6
2 Test	8
2.1 Test med anvendelse af RASP	8
2.2 Konvertering af PKCS12 til JKS	8
3 Anvendelse af TCPmon til test	9

Dette dokument beskriver e-TL kommunikationstest

1 Kommunikationstest

For at sikre en tidlig aftestning af forbindelsen fra eksterne parter til e-TL er der implementeret en række Web Services, til at teste kommunikationen. Der er tale om følgende Web Services

- *KommunikationTest*
Simpel Webservice uden WS-Security og uden WS-ReliableMessaging
- *KommunikationTestWSS*
Webservice med WS-Security men uden WS-ReliableMessaging
- *KommunikationTestRM*
Webservice uden WS-Security men med WS-ReliableMessaging
- *KommunikationTestWSSRM*
Webservice med både WS-Security og WS-ReliableMessaging

Alle disse services kan med følgende URL prefix alt efter hvilket miljø man ønsker at teste imod

Miljø	URL prefix
Udvikler lokal	http://localhost:8081/etlengine/services
Lokal test Århus	http://etlapp01.dk.emea.csc.com:7001/etl-engine/services
Udviklingsmiljø GIS	http://tswbd01.csc.dk/service (via Apache) http://tswbd03.csc.dk:7033/etl-engine/services (direkte til WebLogic)
Internt testmiljø GIS	http://tswbt01.csc.dk/service (via Apache) http://tswbt02.csc.dk/service (via Apache) http://tswapt05.csc.dk:7033/etl-engine/services (direkte til WebLogic) http://tswapt06.csc.dk:7033/etl-engine/services (direkte til WebLogic)
Eksterne aktøreres test	http://<adresse>/service

Eksempelvis benyttes `http://localhost:8081/etlengine/services/KommunikationTest` for at tilgå KommunikationTest lokalt for en udvikler, som har en lokal etl-engine kørende i en Jetty container, og `http://tswbd03.csc.dk:7033/etl-engine/services/KommunikationTest` til at tilgå KommunikationTest i udviklingsmiljøet hos GIS. Husk, at tilgang til miljøet i GIS kræver VPN-opkobling til Domstolsstyrelsens netværk. For de eksterne aktøreres test benyttes `http://<adresse>/service/KommunikationTest` for at tilgå KommunikationTest. Dette går direkte over Internettet. I kommunikationstesten benyttes ikke SSL i første omgang. Alle services i kommunikationstesten vil på sigt være tilgængelige både via `http` og via `https`.

Alle services kræver pt. At der anvendes WS-Addressing headers, hvor konventionen er, at action er `http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/` efterfulgt af navn på test service med lille begyndelsesbogstav. Kald til KommunikationTest benytter altså WS_Addressing action `http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/kommunikationTest`.

Alle services tager en SOAP body med rod-element `KommunikationTest` i namespace `http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/`. Der er ikke yderligere krav til indhold af SOAP body. Følgende er et simpelt eksempel på en besked til kommunikationstest.

```
<tls:KommunikationTest
  xmlns:tls="http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/">
</tls:KommunikationTest>
```

Alle kommunikationstest services returnerer en SOAP body med rod-element `KommunikationTestSvar` i namespace

e-Tinglysningsprojektet

`http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/`. Som sub-element findes elementet `KommunikationTest`, som er en kopi af den forespørgsel, som blev lavet. Herudover kan der også være et sub-element, `CertifikatInfo` i namespace `http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/`, som indeholder information om det certifikat som er blevet anvendt hvis WS-Security benyttes. Følgende er et eksempel på svar på ovenstående, hvis der ikke er anvendt WS-Security

```
<tls:KommunikationTestSvar
  xmlns:tls="http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/">
  <tls:KommunikationTest/>
</tls:KommunikationTestSvar>
```

Hvis der er anvendt WS-Security kan svar se ud som følger

```
<?xml version="1.0" encoding="UTF-8"?>
<tls:KommunikationTestSvar
  xmlns:tls="http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/">
  <tls:KommunikationTest/>
  <tls:CertifikatInfo>
    ...
  </tls:CertifikatInfo>
</tls:KommunikationTestSvar>
```

1.1 KommunikaionTest

Eksempel på forespørgsel incl. (nogle) HTTP headers:

```
POST /etlengine/services/KommunikationTest HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/kommunikationTest"
User-Agent: Axis2
Host: 127.0.0.1:8079
Content-Length: ...

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsa:To>http://localhost:8079/etlengine/services/KommunikationTest</wsa:To>
    <wsa:MessageID>urn:uuid:D892BBB6BCFC2EEDB51182873824717</wsa:MessageID>
    <wsa:Action>http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/kommunikationTest</wsa:Action>
  </soapenv:Header>
  <soapenv:Body>
    <tls:KommunikationTest
      xmlns:tls="http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/"><tls:KommunikationTest>
    </soapenv:Body>
  </soapenv:Envelope>
```

Tilhørende svar incl. (nogle) HTTP headers

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=UTF-8
Server: Jetty(6.0.1)
Content-Length: ...

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsa:Action>http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/kommunikationTestSvar</wsa:Action>
    <wsa:RelatesTo>urn:uuid:D892BBB6BCFC2EEDB51182873824717</wsa:RelatesTo>
  </soapenv:Header>
  <soapenv:Body>
    <tls:KommunikationTestSvar xmlns:tls="http://rep.oio.dk/tinglysning.dk/xml.schema/2008.01.01/">
      <tls:KommunikationTest></tls:KommunikationTest>
    </tls:KommunikationTestSvar>
  </soapenv:Body>
</soapenv:Envelope>
```

1.2 KommunikationTestWSS

Eksempel på forespørgsel incl. (nogle) HTTP headers

```
POST /etlengine/services/KommunikationTestWSS HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "http://rep.oio.dk/tinglysning.dk/xml.wsdl/2008.01.01/kommunikationTestWSS"
User-Agent: Axis2
Host: 127.0.0.1:8079
Content-Length: ...

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      soapenv:mustUnderstand="1">
      <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">
```

e-Tinglysningsprojektet

```

ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="CertId-27310791">
  MIIF....
</wss:BinarySecurityToken>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-27455534">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <ds:Reference URI="#Id-30149247">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <ds:DigestValue>7hi+N9IFXxjMnaMsrMvm9sOXbIg=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>b510...</ds:SignatureValue>
  <ds:KeyInfo Id="KeyId-25528992">
    <wss:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="STRId-6457092">
      <wss:Reference URI="#CertId-27310791"
        Value="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
    </wss:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wss:Security>
<wsa:To>http://localhost:8079/etlengine/services/KommunikationTestWSS</wsa:To>
<wsa:MessageID>urn:uuid:FA6FC352384013F9D81182875884229</wsa:MessageID>
<wsa:Action>http://rep.oio.dk/tinglynsning.dk/xml.wsdl/2008.01.01/kommunikationTestWSS</wsa:Action>
</soapenv:Header>
<soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-30149247">
  <tls:KommunikationTest xmlns:tls="http://rep.oio.dk/tinglynsning.dk/xml.schema/2008.01.01/">
  </tls:KommunikationTest>
</soapenv:Body>
</soapenv:Envelope>

```

Tilhørende svar incl. (nogle) HTTP headers

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=UTF-8
Server: Jetty(6.0.1)
Content-Length: ...

```

```

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wss:Security xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      soapenv:mustUnderstand="1">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-20259911">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference URI="#Id-8135054">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>K1SR7DFcNK9HzEv1BluB7SyBcKI=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>cXjf...</ds:SignatureValue>
        <ds:KeyInfo Id="KeyId-28378327">
          <wss:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
            wsu:Id="STRId-11850652">
            <wss:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
              Value="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">D17Nz+vDmUYXx7ofjo7QzHzZM9w=</wss:KeyIdentifier>
          </wss:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wss:Security>
    <wsa:Action>http://rep.oio.dk/tinglynsning.dk/xml.wsdl/2008.01.01/kommunikationTestWSSvar</wsa:Action>
    <wsa:RelatesTo>urn:uuid:FA6FC352384013F9D81182875884229</wsa:RelatesTo>
  </soapenv:Header>
  <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="Id-8135054">
    <tls:KommunikationTestSvar xmlns:tls="http://rep.oio.dk/tinglynsning.dk/xml.schema/2008.01.01/">
      <tls:KommunikationTest></tls:KommunikationTest>
      <tls:CertifikatInfo>
        <Certificate>
          <Type>X.509</Type>
          <Version>3</Version>
          <Subject>SERIALNUMBER=CVR:26769388-UID:1172691221366 + CN=IT- og Telestyrelsen - OIOSI test virksomhedssignatur, O=IT- og
            Telestyrelsen // CVR:26769388, C=DK</Subject>
          <SerialNumber>1077321566</SerialNumber>
          <Issuer>CN=TDC OCES Systemtest CA II, O=TDC, C=DK</Issuer>
          <ValidFrom>Wed Feb 28 20:14:26 CET 2007</ValidFrom>
          <ValidTo>Sat Feb 28 20:44:26 CET 2009</ValidTo>
        </Certificate>
      </tls:CertifikatInfo>
    </soapenv:Body>
  </soapenv:Envelope>

```

2 Test

2.1 Test med anvendelse af RASP

En mulighed for at kalde kommunikationstest services er at benytte RASP. Det betyder basalt set, at der anvendes Axis2 med Rampart til håndtering af WS-Security og Sandesha til håndtering af WS-ReliableMessaging.

I etl-engine projektet findes test `com.csc.etl.service.KommunikationTest`, som har fire test, som hver kalder en af kommunikationstest services. I denne test benyttes filen `resource/test/service/dynamic_wss_policy.xml` til at initialisere Axis2, Rampart og Sandesha.

2.2 Konvertering af PKCS12 til JKS

Det ser ud til, at den crypto provider, som RASP anvender (`org.apache.ws.security.components.crypto.Merlin`) ikke fungerer med PKCS12 filer fra Windows. Som en del af Jetty findes klassen `org.mortbay.jetty.security.PKCS12Import`, som kan konvertere fra PKCS12 til JKS.

```
> java -cp lib\jetty\jetty-6.0.1.jar org.mortbay.jetty.security.PKCS12Import \
    keys\TestVOCES2.pfx keys\TestVOCES2.jks
```

3 Anvendelse af TCPmon til test

En mulighed for at følge Web Service kommunikationen er at benytte Apache TCPmon. Den kan hentes på <http://ws.apache.org/commons/tcpmon>.