



e-TL System til System kommunikationstest

Version	Dato	Forfatter	Kommentarer	Distribueret til
0.5	22/10-07	Anders Bohn Jespersen	Udgave til workshop 24/10.	
0.6	24/10-07	HGK	Opdateret med beskeder.	
0.9	31/10-07	René Vangsgaard	Opdateret med installationsvejledning	



Indholdsfortegnelse

Formål.....	3
Dette dokument.....	3
Kommunikationstesten.....	3
Afvikling.....	4
Support.....	4
Tidsplan.....	4
Scenarie til kommunikationstest.....	5
Generelt	5
Hvad valideres?.....	5
Hvad er ikke med?.....	5
Scenarie 1 - fra eksternt system til e-TL.....	5
Scenarie 2 - fra e-TL til eksternt system.....	6
Testkit.....	7
RASP-klient.....	7
Rasp-service.....	7



Formål

Dette dokument

Dette dokument beskriver kommunikationstest for den servicesnitflade e-TL stiller til rådighed for de eksterne system-til-system brugere.

Dokumentet skal ses i sammenhæng med den oprindelige løsningsbeskrivelse for e-TL (fx afsnit 4 fra 1. juni 2007) og beskrivelsen af RASP-integration, *S2S grænseflader*.

Kommunikationstesten

Har til formål at få afprøvet transporten af meddelelser til og fra e-TL. Indholdet i meddelelser er rent 'hello world'.

Testen er to-vejs, i det eksterne systemer skal kunne kalde e-TL, og e-TL skal kunne kalde de eksterne systemer.

Det betyder, at en komplet kommunikationstest involverer opsætning af en RASP-service i det eksterne system.

I forbindelse med testen udleveres et testkit bestående af en RASP-klient og RASP-service. Testkittet er udviklet til Java.



Afvikling

Support

Generel driftsinformation sendes via T&T-mailinglisten og er tilgængeligt via portalen på: <http://portal.csc.com>

CSC leverer support dels igennem T&T-mødet hver torsdag.

Er der mangler i dette informationsmateriale eller yderligere spørgsmål kan disse stilles på mail: tdp@lenio.dk

Spørgsmål til selve afvikling (fx 'hvorfor får jeg ingen svar retur fra e-TL') kan stille på mail: tdp@lenio.dk

I tilfælde af blokerende fejl eller mangler, kontakt telefonnummer: 2728 1535

Tidsplan

Kommunikationstesten med 100% RASP kan køres fra eksterne systemer fra 1/11, 2007.

Endpoint fremgår af WSDL for kommunikationsteste.



Scenarie til kommunikationsstest

Generelt

Der anvendes OCES-test-certifikater udstedt af TDC til alle kald. På e-TL-siden er serveren konfigureret til at benytte :

CN=NemHandel Test 2 + SERIALNUMBER=CVR:26769388-DID:00000002, O=IT- og Telestyrelsen // CVR:26769388, C=DK"

Samme certifikat/privat nøgle benyttes til signatur på statusbeskeder.

Som klient kan benyttes et vilkårligt OCES-test-certifikat udstedt af TDC. Det klientcertifikat der benyttes til signering, vil efterfølgende blive brugt når statusbeskeder sendes retur.

Certifikater og nøgler kan rekvireres hos TDC, eller hos e-TL-teamet.

Hvad valideres?

I kommunikationstesten valideres messages ikke, men alt vedrørende transport og RASP kontrolleres.

Dvs transportsignering og -kryptering indgår.

Hvad er ikke med?

Indhold (anmeldelser, søgninger) indgår ikke. Der er følgelig heller ingen signaturer i payload. Det samme certifikat kan benyttes af flere aktører, og der er altså ingen skelnen mellem eksterne systemer (system2system-ordningen).

Disse ting tilgår tidligst i meddelelsestesten.

Scenarie 1 - fra eksternt system til e-TL

Eksternt system sender en simpel tekstbesked til test-service og får synkront et svar retur af samme slags med fremsendte tekst indlejret.

```
<Besked>  
  <Tekst>Tekst som kommer retur</Tekst>  
</Besked>
```

Svaret der bliver genereret vil se således ud :



```
<etl:Svar>
  <etl:DatoLoebenummer>2007-10-23T21:46:33-1</etl:DatoLoebenummer>
  <etl:ReturTekst>Tekst som kommer retur</etl:ReturTekst>
</etl:Svar>
```

I svaret indgår et simuleret 'DatoLoebenummer'.

Scenarie 2 – fra e-TL til eksternt system

Den modsatte retning kan aftestes ved at tilføje en SvarUrl til indsendte besked. SvarUrl skal indeholde URL til endpoint.

```
<Besked>
  <SvarUrl>http://321.321.321.321/status/services/Status</SvarUrl>
  <Tekst>Tekst som kommer retur</Tekst>
</Besked>
```

Dette afføder at e-TL - udover det synkron svar som i scenarie 1 - efter kort tid foretager et separat kald retur til angivne endpoint-URL, med fremsendte tekst indlejret.

```
<etl:Status>
  <etl:MessageIdentifier>
    UniqueId-1193169013650087000
  </etl:MessageIdentifier>
  <etl:DatoLoebenummer>2007-10-23T21:50:15-1</etl:DatoLoebenummer>
  <etl:Tekst>Dette er en status tekst</etl:Tekst>
</etl:Status>
```

I det asynkrone 'Status' kald indgår felterne:

MessageIdentifier: som er den unikke identifikation, som Aktør satte på den fremsendte 'Besked', jævnfør RASP-specifikation TODO.

DataLoebenummer: som simulerer e-TL's registrering af beskeden.



Testkit

Testkittet består af skeletkode til at udvikle en RASP-klient (til afsendelse af beskeder), og en RASP-service (til modtagelse af statusbeskeder).

Forudsætninger:

- Java 1.4.2 eller nyere.
- Eclipse 3.3 eller nyere, se <http://www.eclipse.org>.
- Ant 1.7 eller nyere, se <http://ant.apache.org>.
- Servlet-container version 2.3 eller nyere. Se evt. <http://tomcat.apache.org>.

RASP-klient

RASP-klienten er et stand-alone Java-program. I testkittet er inkluderet konfiguration således programmet kan køres fra Eclipse.

For at kigge i og køre programmet fra Eclipse følg nedenstående instruktioner:

1. Start Eclipse. Vælg et vilkårligt Workspace.
2. File => Import... => General => Existing Projects into Workspace. Klik Next.
3. I feltet 'Select root directory' indsættes folderen /prof-rasp-client. Kontrollér, at projekt prof-rasp-client er markeret i feltet Projects. Klik Finish.

Udførbár klasse: `dk.firma.klient.webservice.BeskedClient` – ligger i filen `/src/dk/firma/klient/webservice/BeskedClient.java`.

Mulige tilpasninger findes i toppen af klassen `BeskedClient`, kig efter teksten `CHANGEABLE`.

Rasp-service

RASP-servicen er et Java web-projekt (Servlet 2.3). RASP-servicen giver kontrollen videre til klassen `dk.firma.webservice.StatusService`, hvor forretningslogikken på et statussvar skal implementeres. Klassen ligger i filen `/src/dk.firma.webservice.StatusService.java`.



e-TL System til System kommunikationstest

For at importere og kigge i programmet fra Eclipse følg nedenstående instruktioner:

4. Start Eclipse. Vælg et vilkårligt Workspace.
5. File => Import... => General => Existing Projects into Workspace. Klik Next.
6. I feltet 'Select root directory' indsættes folderen /prof-service. Kontrollér, at projekt prof-service er markeret i feltet Projects. Klik Finish.

Servicen bygges ved hjælp af Ant med kommandoen:

```
ant war
```

Bemærk, kommandoen skal udføres i folderen /prof-service.

Nu kan filen status.war deployeres i en servlet-container¹.

Husk, at aktivere statussvar i BeskedClient ved at sætte variabelen svarHostOgPort til not-null.

¹ En WAR-fil deployeres på Tomcat ved at placere filen i /tomcat-home/webapps/. Når Tomcat starter op, deployeres filen automatisk.