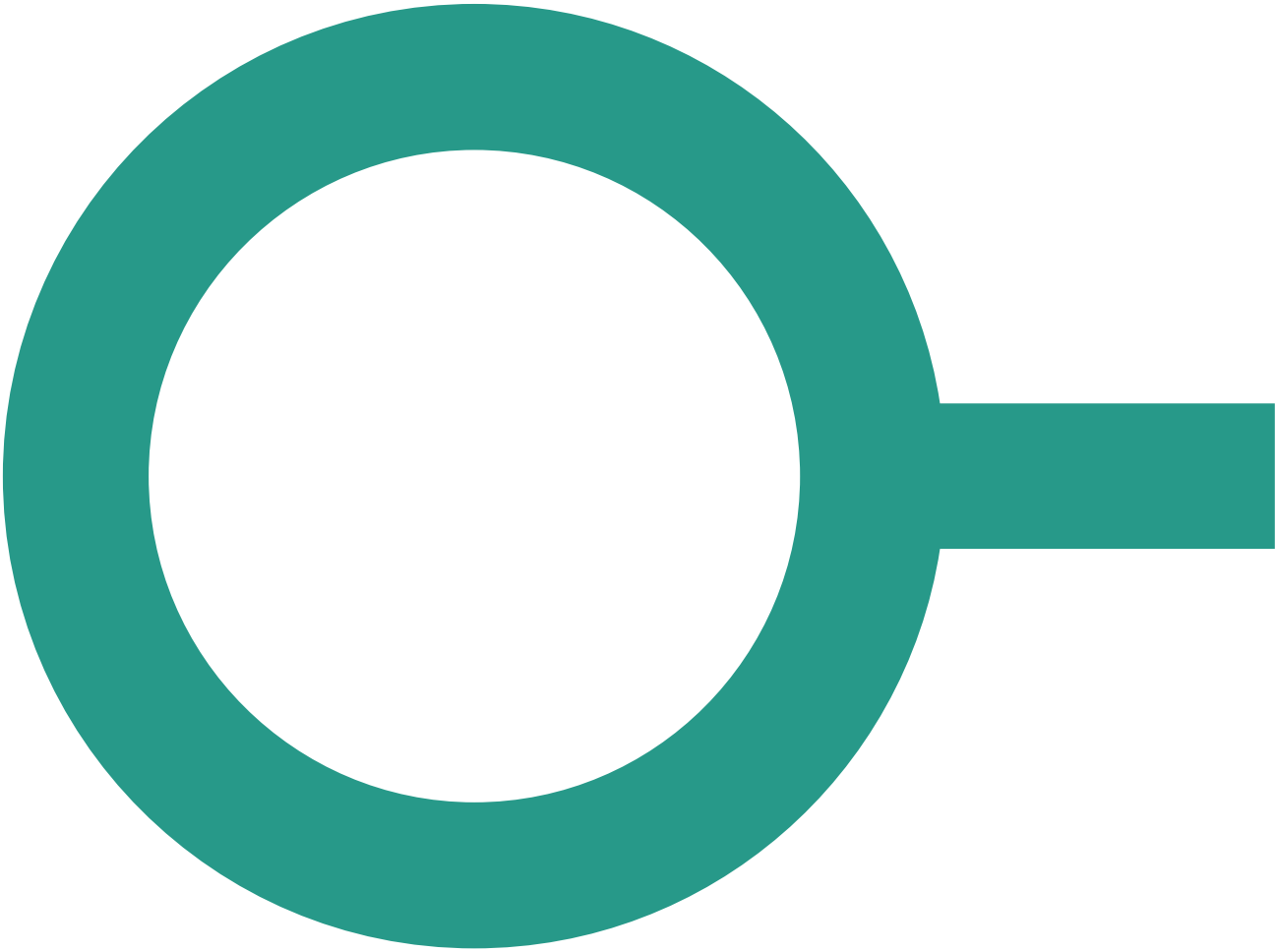


PROGRAMMERS GUIDE

SIKKERHED I DEN FÆLLESKOMMUNALE INFRASTRUKTUR - SÅDAN KALDER DU SERVICES UDSTILLET VIA SERVICEPLATFORMEN

Version 1.1

KOMBIT

Kommunernes it-fællesskab

Versionshistorik

Version	Dato	Kommentar
0.1	27.09.2018	Intiel udgave
0.2	27.01.2019	Tilføjet afsnit 6 Authority context samt XML payload eksempel.
1.0	22.03.2019	Udgivet
1.1	18.02.2020	Opdateret til Digitaliseringskataloget

Indholdsfortegnelse

1 INDLEDNING

Serviceplatformen er en del af den fælleskommunale infrastruktur, og understøtter de sikkerhedsmodeller, der indgår i infrastrukturen. Det betyder, at services udstillet via Serviceplatformen, understøtter én eller flere sikkerhedsmodeller.

Der er p.t. 3 forskellige sikkerhedsmodeller, som går i den fælleskommunale infrastruktur.

I dette dokument anvendes de navne på sikkerhedsmodellerne, som de er implementeret under på Serviceplatformen, dvs InvocationContext, AuthorityContext og Security Token.

Sikkerhedsmodellerne omtales i de overordnede beskrivelser af sikkerhedsmodellen som

- Simpel = InvocationContext
- Simpel fælleskommunal = AuthorityContext
- Fælleskommunal = Security Token

2 LÆSEVEJLEDNING

Denne vejledning giver et overblik over de tre forskellige sikkerhedsmodeller, som anvendes i den fælleskommunale infrastruktur. Herudover beskriver Programmers Guide, hvordan du kalder services på Serviceplatformen med de forskellige sikkerhedsmodeller.

De tre sikkerhedsmodeller bliver gennemgået enkeltvis og giver et overblik over, hvilke aktiviteter, der skal til for, at du som leverandør kan implementere dem succesfuldt. Herudover beskrives de enkelte aktiviteter trin for trin.Overblikket kan du bruge til at opnå en bedre planlægning og forståelse af opgavens omfang og de nødvendige trin i implementeringen – både for dig som it-leverandør, men også de trin som kommunen skal udfører. Der er udarbejdet en tjekliste til hver af sikkerhedsmodellerne, som du kan bruge til at understøtte din implementering af den konkretesikkerhedsmodel.

Den detaljerede gennemgang er primært målrettet dig, som skal udføre den tekniske implementering af integrationen til Serviceplatformen. Derfor er der lagt vægt på de tekniske elementer og mindre fokus på baggrunden for sikkerhedsmodellerne. Til hver sikkerhedsmodel er der knyttet kodeeksempler

Den tekniske basisviden, som vi forudsætter er forskellige for de 3 sikkerhedsmodeller, er beskrevet i det relevante afsnit i forhold til den konkrete sikkerhedsmodel.

Du finder links til den relevante dokumentation i referencelisten bagerst i denne vejledning.

3 SIKKERHEDSMODELLER I DEN FÆLLESKOMMUNALE INFRASTRUKTUR

3.1 De 3 sikkerhedsmodeller

Der er p.t. tre sikkerhedsmodeller understøttet af Serviceplatformen: InvocationContext, AuthorityContext og Security Token sikkerhed

- **Invocation Context** er baseret på certifikatsikkerhed, dvs anvendersystemet etablerer en såkaldt 2-vejs SSL forbindelse, hvor anvendersystemet identificerer sig med et klient certifikat. I denne model er der ingen reference til en myndighed, og adgang gives alene på baggrund af kombinationen af serviceaftale og certifikat. I kaldet mod Serviceplatformen skal man angive UUID'et på den serviceaftale man anvender.

- **Authority Context** er ligeledes baseret på certifikatsikkerhed, men kaldet foregår altid i kontekst af en bestemt myndighed, og CVR nummeret på myndigheden angives i kaldet.

- **Security Token** er som Authority Context myndighedsspecifik, og kombinerer certifikatsikkerhed med et SAML token, der indeholder roller, dataafgrænsninger og myndighedskontekst.

3.2 Serviceaftaler

Alle tre sikkerhedsmodeller er suppleret med Serviceaftaler, som leverandøren af et it-system og en kommune indgår i STS Administrationsmodul

En serviceaftale er det der giver dig adgang til en kommunes data. Det er en aftale mellem dig og kommunen, om at et konkret it-system må få adgang til konkrete data. Servieaftalen er samtidig kommunes instruks til KOMBIT fra kommune om, at der må udleveres data til det pågældende it-system.

4 KOM GODT I GANG

Før du går i gang med at integrere til services udstillet på Serviceplatformen, er der nogle ting du skal sikre dig er på plads inden.

4.1 Oprettelse som leverandør

For at kunne kalde webservices udstillet via Serviceplatformen, skal din organsiation være oprettet som leverandørorganisation på Serviceplatformen og i STS Administrationsmodulet. Du opretter din organisation ved at kontakte Helpdesk på tlf: *70 11 15 39* eller ved at skrive til *helpdesk@serviceplatformen.dk*, og bede om en oprettelsesblanket for leverandører.

Du vil blive bedt om at udfylde RID-nr, på den medarbejder som skal oprettes som administrator. RID-nr. er det ID, som findes i medarbejderens medarbejdercertifikat (MOCES).

Når blanketten er udfyldt skal du indende den til Helpdesk.

Når din oprettelse er godkendt, bliver din organisation oprettet i både STS Administrationsmodulet og Serviceplatformens administrationsmodul. Du kan læse mere om funktionaliteten i STS administrationsmodulet i vejledningen [ADMINISTRATION]

For at logge på STS Administrationsmodulet skal du tildele medarbejdere i din organisation, som skal anvende STS Administrationsmodulet de nødvendige rettigheder i NemLogin. [NEMLOGIN]

I forhold til login i Serviceplatformens administrationsmodul, bliver den første administrator oprettet samtidig med oprettelsen af din organisation. Administratoreren kan efterfølgende selv oprette brugere med de nødvendige roller i Serviceplatformens administrationsmodul [ADMINISTRATION]

4.2 Opret dit it-system i STS Administrationsmodulet

Du skal oprette dit it-system som anvendersystem i STS Administrationsmodulet, når du skal kalde services på Serviceplatformen. I forbindelse med oprettelsen af dit it-system skal du anvende et certifkat. Det kan du med fordel bestille hos DanID inden du påbegynder oprettelsen.

Vi anbefaler at du anvender et FOCES-certifkat. I ekstern test kan du anvende et test FOCES-certifikat eller produktions FOCES-certifikat til dit system. I produktionmiljøet er det et krav at du anvender et produktions FOCES-certifikat. Vi anbefaler at du anvender forskellige certifikater i test og produktion.

Læs mere om oprettelse af it-systemer i STS Administrationsmodulet i vejledningen [ADMINISTRATION]

4.3 Godkendelse af serviceaftaler

Før du kan få adgang til data og kalde en service skal du have en godkendt Serviceaftale på den eller de konkrete services du ønsker at få adgang til. Det er en aftale mellem dig og kommunen om at et konkret it-system må få adgang til konkrete data. Det er samtidig kommunes instruks til KOMBIT om, at der må udleveres data.

I det eksterne test system kan Serviceplatformens heldesk godkende på vegne af kommunen. Du skal blot sende en mail til helpdesk@serviceplatformen.dk.

I produktion er det naturligvis kun kommunens aftaleadministrator i STS Administrationsmodulet, som har ret til at godkende serviceaftaler.

4.4 Digtaliseringskataloget

Digitaliseringskataloget giver dig overblik over de services, der udstilles via den fælleskommunale infrastruktur. Her kan du læse om de enkelte services og finde links til relevante vejledninger og dokumentaion. I Servicekataloget kan du også se, hvilke af de 3 sikkerhedsmodeller en konkret service er udstillet med.

5 INVOCATION CONTEXT

Invocation Context er den simple sikkerhedsmodel, hvor det alene er muligt at styre hvorvidt et anvendersystem må kalde en service eller ej. Der er ingen myndighedskontekst, ingen afgrænsning af data, og intet rollebegreb i denne sikkerhedsmodel. Adgang til en service styres udelukkende ud fra om der er en godkendt serviceaftale mellem et anvendersystem, en myndighed og en service.

5.1 Sådan fungerer Invocation Context

I kaldet mod Serviceplatformen udfyldes den del af SOAP konvolutten der vedrører InvocationContext. Her skal du unikt identificerer den serviceaftale du kalder med, ved angivelse af en række UUID'er. UUID'er kan du læse mere om i afsnit 5.3.

Selve forbindelsen etableres som en såkaldt 2-vejs SSL forbindelse, hvor du bruger dit FOCES-certifikat som klient- certifikat. Du skal anvende det certfikat, du har registreret på dit it-system i STS-Administration.

Serviceplatformen validerer SSL-forbindelsen, og dermed klient-certifikatet, og såfremt certifikat og UUID'er stemmer overens, og der serviceaftalen indeholder den konkrete service, får du adgang til at kalde servicen.

5.2 Step 1: test med demo-service

Du kan med fordel starte med at teste med demoklienten før du påbegynder at din egentlige test af en service.

5.2.1 Hent Kodebasen til demo-service

Til Demo-Service er der udstillet eksempelkode, som viser hvordan du trækker et token i hhv. Java og .NET, og denne kode kan hentes her

https://github.com/Serviceplatformen/demoservice-client-java

https://github.com/Serviceplatformen/demoservice-client-net

Kodebasen er født med et FOCES certifikat, der er registreret på et it-system i test-miljøet, og der er indgået en serviceaftalen med en enkelt test-service, så koden kan afvikles som den er, med det resultatet der hentes et token fra Security Token Servicen, og der gennemføres et kald mod test-servicen på Serviceplatformen.

Nedenfor får du nogle praktiske detaljer om de to kodeeksempler

5.2.2 Java eksempel

Kodebasen til Jave indeholder 2 eksempler, hvor det ene er til den gamle sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den gamle context-baserede sikkerhedsmodel, som du finder i folderen:

```
demoservice-context-client
```

De to primære frameworks der anvendes i eksempelkoden er Spring og CXF , og koden bygges med værktøjet Maven.

Hvis du ikke har erfaring med at bruge CXF frameworket, kan man med fordel læse dokumentationen til den CXF-baserede referencekode som Digitaliseringsstyrelsen vedligeholder CFX.

Det er i resource-filen cxf.xml at det meste af den relevante konfiguration kan findes, og her konfigureres bl.a. kaldet til servicen og etableringen af 2-vejs SSL forbindelsen til Serviceplatformen. De specifikke parametre til InvocationContext konfigureres i resource-filen invocationcontext.properties.

5.2.3 .NET eksempel

Kodebasen til .NET indeholder 2 eksempler, hvor det ene er til den context-basedere sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den context-baserede sikkerhedsmodel, som du finder i folderen:

```
DemoClient
```

Kodebasen er skrevet i C#, og baserer sig på Microsofts eget framework til webservices (WCF). Endvidere er der inddraget et framework til at håndtere Liberty Basic SOAP Binding profilen, udarbejdet af Digitaliseringstyrelsen [OIO IDWS 3.0].

Den version af frameworket der indgår i eksempelkoden har dog nogle få ændringer, så den er kompatibel med de krypto-algoritmer som KOMBIT anvender, samt at den understøtter etableringen af en 2-vejs SSL forbindelse.

Kodebasen anvender XML-baseret konfiguration af WCF, og hvis du i sin egen kode ønsker at konfigurere WCF via kode og ikke XML, så er det vigtigt, at du får alt fra App.config med fra eksempelkoden, da den indeholder konfiguration af OIO IDWS frameworket.

Klienten kører umiddelbart, såfremt man følger instruksen og installerer:

- .NETFramework 4.5.2.
- Microsoft Build Tools 2015 (Hvis du ikke ønsker at installere Visual Studio, men blot vil afvikle eksempelklienten)

  og eksekverer run.bat fra en kommandoprompt for at gennemføre kald til demo-service.

5.3 Step 2: Gennemfør test i ekstern test mod konkret webservice

Når du kalder en service, som understøtter InvocationContext, skal du sende en række informationer med i XML kaldet, for at identificere dit it-system, og blive godkendt adgang til at kalde webservicen. Flere af disse parametre bliver dannet i forbindelse med oprettelse af en Serviceaftale.

5.3.1 Oprette en Serviceaftale på den konkrete service i ekstern test

Nu skal du oprette en serviceaftale på den service, du ønsker at teste forretningsmæssigt i forhold til din løsning.

- Log på STS Administrationsmodulet i ekstern test
- Opret en serviceaftale på den konkrete service du ønsker at anvende
- Få helpdesk til at godkende serviceaftalen ved at sende en e-mail besked til helpdesk@serviceplatformen.dk
- Noter følgende parameter fra Serviceaftalen
  - ServiceaftaleUUID
  - UserSystemUUID

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen. Du kan læse mere om hvordan du opretter serviceaftaler i vejledningen til STS Administration [ADMINISTRATION].

Du skal anvende følgende informationer i InvocationContext i dit SOAP XML payload

Feltnavn	Værdi
ServiceAgreementUUID	Aftale UUID fra Serviceplatformen under ”Serviceaftaler”
UserSystemUUID	System UUID fra Serviceplatformen under ”Serviceaftaler”
UserUUID	Kommune UUID fra Serviceplatformen under ”Serviceaftaler”
ServiceUUID	Service UUID fra Serviceplatformen under ”Find service” eller ”Serviceaftaler”
OnBehalfOfUser	Valgfrit felt som kan indeholde f.eks. brugerkontonavn i anvendersystemet. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.
CallersServiceCallIdentifier	Valgfrit felt som kan indeholde f.eks. internt sagsnummer. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.
AccountingInfo	Valgfrit felt som kunne indeholde f.eks. faktureringsinformation. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.

Du finder ServiceAgreementUUID og UserSystemUUID i STS Administrationsmodulet ved at åbne din serviceaftale og du kan finde en liste over kommunernes UserUUID i KDIs dokumentbibliotek [USERUUID] og ServiceUUID i Serviceplatformens administrationsmodul.

5.4 Step 3: Ibrugtagning i produktion

5.4.1 Registrer it-system i produktion

Før du kan gennemføre kald i produktion skal du registre dit it-system i produktion.

Du skal sørge for, at

- Oprette din organisation i STS Administration i produktion
- Registrere dit it-system i STS Administration
- Tilknytte et gyldigt FOCES certifikat med SAML metadata til dit it-system

Hvis din organisation endnu ikke er oprettet på produktionmiljøet skal du kontakte helpdesk@serviceplatformen.dk

5.4.2 Opret en Serviceaftale på konkret service i produktion

- Log på STS Administration
- Opret en serviceaftale på demo-service
- Få kommunen* til at godkende serviceaftalen
- Noter følgende parameter fra Serviceaftalen
    - CVR-nr på myndighed
    - Service EntityId
    - Service Endpoint

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen.

Husk du kan se hvordan du anmoder om serviceaftale i vejledning fra tidligere.

6 AUTHORITY CONTEXT

AuthorityContext er en implementation af den Simple fælleskommunale sikkerhedsmodel. Modellen minder om InvocationContext, men kald til servicen sker altid i kontekst af én bestemt myndighed, der angives i kaldet. Envidere skal hver myndighed (som der skal kaldes i kontekst af) godkende aftalen forinden.

6.1 Sådan fungerer AuthorityContext (Myndighedskontekst)

I kaldet mod Serviceplatformen udfyldes den del af SOAP konvolutten der vedrører AuthorityContext. Her skal man angive CVR nummeret på den myndighed, man kalder på vegne af, i feltet MunicipalityCVR.

Selve forbindelsen etableres som en såkaldt 2-vejs SSL forbindelse, hvor man bruger sit FOCES certifikat (det som er registreret for ens eget IT-system), som klient certifikat.

Serviceplatformen validerer SSL forbindelsen, og dermed klient certifikatet, og hvis certifikat og CVR matcher en registreret serviceaftale på den konkrete service, så gives der adgang til at kalde servicen.

6.2 Step 1: test med demo-service

Du kan med fordel starte med at teste med demoklienten før du påbegynder at din egentlige test af en service.

6.2.1 Hent Kodebasen til demo-service

Til Demo-Service er der udstillet eksempelkode, som viser hvordan du trækker et token i hhv. Java og .NET, og denne kode kan hentes her

https://github.com/Serviceplatformen/demoservice-client-java

https://github.com/Serviceplatformen/demoservice-client-net

Kodebasen er født med et FOCES certifikat, der er registreret på et it-system i test-miljøet, og der er indgået en serviceaftalen med en enkelt test-service, så koden kan afvikles som den er, med det resultatet der hentes et token fra Security Token Servicen, og der gennemføres et kald mod test-servicen på Serviceplatformen.

Nedenfor får du nogle praktiske detaljer om de to kodeeksempler

6.2.2 Java eksempel

Kodebasen til Jave indeholder 2 eksempler, hvor det ene er til den gamle sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den gamle context-baserede sikkerhedsmodel, som du finder i folderen:

```
demoservice-context-client
```

De to primære frameworks der anvendes i eksempelkoden er Spring og CXF , og koden bygges med værktøjet Maven.

Hvis du ikke har erfaring med at bruge CXF frameworket, kan man med fordel læse dokumentationen til den CXF-baserede referencekode som Digitaliseringsstyrelsen vedligeholder her.

Det er i resource-filen cxf.xml at det meste af den relevante konfiguration kan findes, og her konfigureres bl.a. kaldet til servicen og etableringen af 2-vejs SSL forbindelsen til Serviceplatformen. De specifikke parametre til InvocationContext konfigureres i resource-filen invocationcontext.properties.

6.2.3 .NET eksempel

Kodebasen til .Net indeholder ligeledes 2 eksempler, hvor det ene er til den gamle sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den gamle context-baserede sikkerhedsmodel, som du finder i folderen:

```
DemoClient
```

Kodebasen er skrevet i C#, og baserer sig på Microsofts eget framework til webservices (WCF).

Klienten kører umiddelbart, såfremt man installerer følgende og følger instruksen fra GitHub:

- .NETFramework 4.5.2.
- Microsoft Build Tools 2015 (Hvis du ikke ønsker at installere Visual Studio, men blot vil afvikle eksempelklienten)

  og eksekverer run.bat fra en kommandoprompt for at gennemføre kald til demo-service.

6.3 Step 2: Gennemfør test i ekstern test mod konkret webservice

Når du kalder en service, som understøtter AuthorityContext, skal du sende en række informationer med i XML kaldet, for at identificere dig selv, og blive godkendt adgang til at kalde webservicen. Flere af disse parametre bliver dannet i forbindelse med oprettelse af en Serviceaftale.

6.3.1 Oprette en Serviceaftale på den konkrete service i ekstern test

Nu skal du oprette en serviceaftale på den service, du ønsker at teste forretningsmæssigt i forhold til din løsning.

- Log på STS Administrationsmodulet i ekstern test
- Opret en serviceaftale på den konkrete service du ønsker at anvende
- Få helpdesk til at godkende serviceaftalen ved at sende en e-mail besked til helpdesk@serviceplatformen.dk
- Noter CVR nummeret for myndigheden fra Serviceaftalen

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen. Du kan læse mere om hvordan du opretter serviceaftaler i vejledningen til STS Administration [ADMINISTRATION].

Du skal anvende følgende informationer i **AuthorityContext** i dit SOAP XML payload

Feltnavn	Værdi
MunicipalityCVR	Påkrævet felt som udfyldes med kommunens CVR nummer. Kan findes på serviceaftalen.

Du kan anvende følgende informationer i **CallContext** i dit SOAP XML payload hvis du har brug for ekstra informationer ved fejlfinding.

Feltnavn	Værdi
OnBehalfOfUser	Valgfrit felt som kan indeholde f.eks. brugerkontonavn i anvendersystemet. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.
CallersServiceCallIdentifier	Valgfrit felt som kan indeholde f.eks. internt sagsnummer. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.
AccountingInfo	Valgfrit felt som kunne indeholde f.eks. faktureringsinformation. Gemmes i loggen på Serviceplatformen, så det kan bruges i forbindelse med support af anvenderne og IT-systemudviklerne.

6.4 Step 3: Ibrugtagning i produktion

6.4.1 Registrer it-system i produktion

Før du kan gennemføre kald i produktion skal du registre dit it-system i produktion.

Du skal sørge for, at

- Oprette din organisation i STS Administration i produktion
- Registrere dit it-system i STS Administration
- Tilknytte et gyldigt FOCES certifikat med SAML metadata til dit it-system

Hvis din organisation endnu ikke er oprettet på produktionmiljøet skal du kontakte helpdesk@serviceplatformen.dk

6.4.2 Opret en Serviceaftale på konkret service i produktion

- Log på STS Administration
- Opret en serviceaftale på demo-service
- Få kommunen* til at godkende serviceaftalen
- Noter følgende parameter fra Serviceaftalen
    - CVR-nr på myndighed
    - Service Endpoint

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen.

7 SECURITY TOKEN

Security Token er en implementation af den Fælleskommunale sikkerhedsmodel, hvor kald altid sker i kontekst af en myndighed, og anvendersystemet er tildelt specifikke roller og dataafgrænsninger som anvendersystemet må agerere på.

Implementationen introducerer en yderligere komponent, der anvendes i kald-mønsteret mod Serviceplatformen. Denne komponent hedder Security Token Servicen (STS), og udsteder SAML tokens, der indeholder anvendersystemets rettigheder, og anvendes i kald mod Serviceplatformen.

Security Token Servicen giver mulighed for en mere finkornet kontrol over systemers adgang til kommunernes data, idet det er muligt at styre hvilke systemer, der må tilgå data og præcist, hvilke data der må tilgås samt hvilke handlinger, der må udføres på dem.

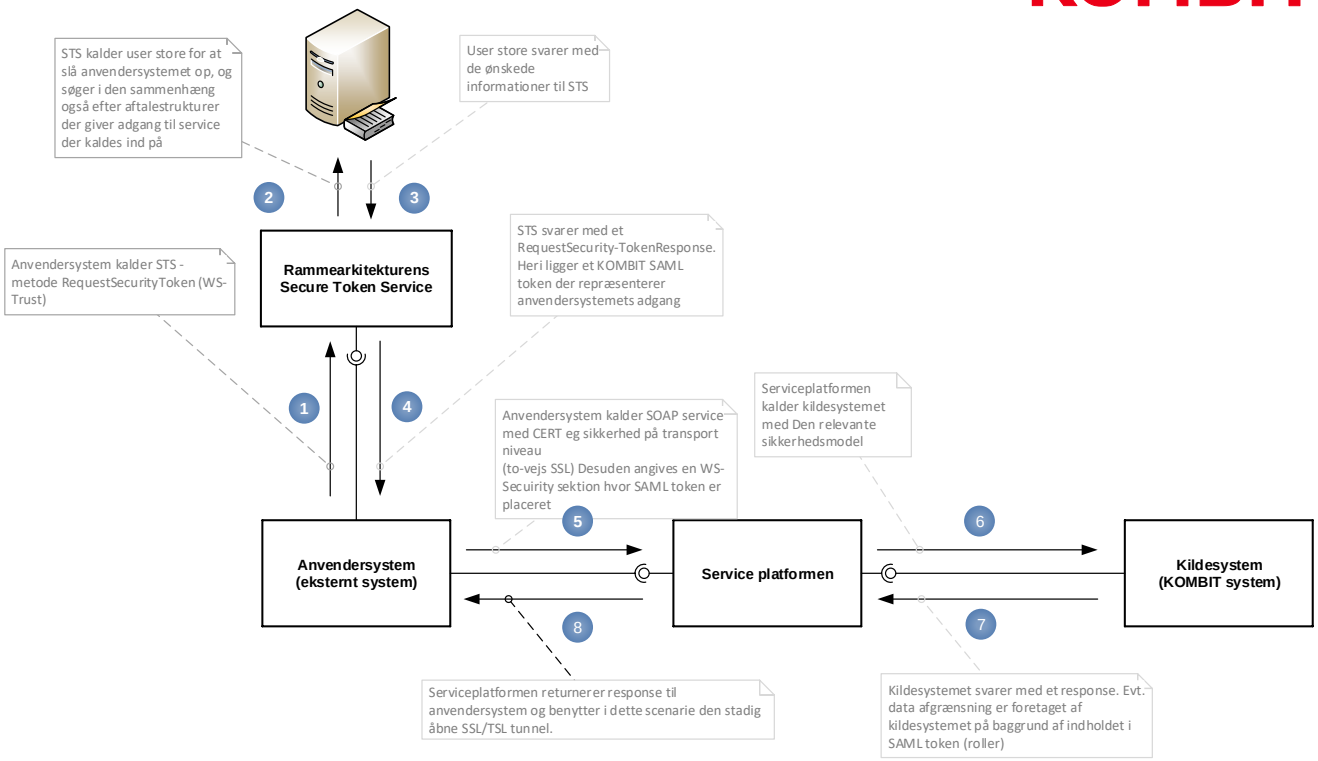
Det giver en bedre sikkerhed for både leverandørerne af kommunale it-løsninger og kommunerne i og med, at det mindsker risikoen for at leverandøren af it-løsningen, har afgang til mere data end der er lovhjemmel til.

Dette kapitel gennemgår, hvordan du som leverandør af en kommunal it-løsning anvender Security Token Services, til at tilgå data via den fælleskommunale infrastruktur.

7.1 Sådan fungerer token-sikkerhed

Den sikkerhedsmodel der anvendes på Serviceplatformen fungerer overordnet på følgende måde

- It-systemet kalder Security Token Servicen for at hente et token til den service på Serviceplatformen som it-systemet ønsker at kalde.
- Et token er service-specifikt, så hvis der skal kaldes flere services, skal der hentes flere tokens; ét token for hver service.
- Tokens har en gyldighedsperiode, og kan genbruges indenfor denne gyldighedsperiode, så man skal ikke trække et nyt token til en service, før ens nuværende token er tæt på at udløbe. Gyldigheden er pt 8 timer, og udløbstidspunktet står i det udstedte token.
- It-systemet etablerer en sikker 2-vejs SSL forbindelse til Serviceplatformen.
- It-systemet generer et SOAP payload, der er struktureret jf Liberty Basic SOAP Binding profilen, og sender det til Serviceplatformen, og har det udstedte token indlejret.
- Serviceplatformen validerer payloadet, og sender et svar tilbage til It-systemet.



Processen med at sikre sig et token hos STS via et WS-Trust kald er beskrevet i dokumentet **Snitfladebeskrivelse-STS-OIO WS Trust.pdf**, og vil ikke blive beskrevet yderligere her.

I kaldet til Serviceplatformen skal det token, som er modtaget fra STS, placeres i en WS-Security header sektion. Dette er udtrykt i en WS-SecurityPolicy WSDL fil, som refereres i WSDL filen for de enkelte services, der anvender denne sikkerhedsmodel. Derudover er det påkrævet at medsende WS-Addressing felter, signere de enkelte SOAP headers og SOAP body og medsende timestamps. Yderligere specifikation af kravene til benyttelse af KOMBIT Token sikkerhedsmodellen kan findes i specifikationen for ”Liberty Basic SOAP Binding Version 1.0 Final” på Project Libertys hjemmeside[1].

Da SOAP headers skal signeres både ved kald til Serviceplatformen, men også i svaret, vil det være nødvendigt at konfigurere trust til Serviceplatformens offentlige certifikat. Kontakt Serviceplatformens Helpdesk ved at skrive til helpdesk@serviceplatformen.dk for fremsendelse af certifikatet.

Før du kan komme i gang med at anvende Security Token Service er der en række ting, du skal igennem. Du skal altid have gennemført test af din løsnings integration mod infrastrukturen, før du går i produktion med din løsning. Din test gennemfører du i det eksterne testmiljø - exttest.

Nedenfor har vi beskrevet hvordan du kommer godt i gang med token-sikkerhed

7.2 Step 1: Test med demo-service

Du kan med fordel starte med at teste mod demoklienten før du påbegynder at din egentlige test af en service.

7.2.1 Hent Kodebasen til demo-service

[1] http://www.projectliberty.org/resource_center/specifications/liberty_basic_soap_binding_version_1_0_final

Til Demo-Service er der udstillet eksempelkode, som viser hvordan du trækker et token i hhv. Java og .NET, og denne kode kan hentes her

https://github.com/Serviceplatformen/demoservice-client-java

https://github.com/Serviceplatformen/demoservice-client-net

Kodebasen er født med et FOCES certifikat, der er registreret på et it-system i test-miljøet, og der er indgået en serviceaftalen med en enkelt test-service, så koden kan afvikles som den er, med det resultatet der hentes et token fra Security Token Servicen, og der gennemføres et kald mod test-servicen på Serviceplatformen.

Nedenfor får du nogle praktiske detaljer om de to kodeeksempler

7.2.2 Java eksempel

Kodebasen til Jave indeholder 2 eksempler, hvor det ene er til den gamle sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den token-baserede sikkerhedsmodel, som du finder i folderen:

```
demoservice-token-client
```

De to primære frameworks der anvendes i eksempelkoden er Spring og CXF , og koden bygges med værktøjet Maven.

Hvis du ikke har erfaring med at bruge CXF frameworket, kan man med fordel læse dokumentationen til den CXF-baserede referencekode som Digitaliseringsstyrelsen vedligeholder her.

Det er i resource-filen cxf.xml at det meste af den relevante konfiguration kan findes, og her konfigureres bl.a. kaldet til servicen, kaldet til Security Token Servicen, og etableringen af 2-vejs SSL forbindelsen til Serviceplatformen.

7.2.3 .NET eksempel

Kodebasen til .Net indeholder 2 eksempler, hvor det ene er til den gamle sikkerhedsmodel på Serviceplatformen, og den anden er til den token-baserede sikkerhedsmodel. Du skal anvende koden til den token-baserede sikkerhedsmodel, som du finder i folderen:

```
DemoTokenClient
```

Kodebasen er skrevet i C#, og baserer sig på Microsofts eget framework til weservices (WCF). Endvidere er der inddraget et framework til at håndtere Liberty Basic SOAP Binding profilen, udarbejdet af Digitaliseringstyrelsen [OIO IDWS 3.0].

https://www.digitaliser.dk/resource/3949880

Den version af frameworket der indgår i eksempelkoden har dog nogle få ændringer, så den er kompatibel med de krypto-algoritmer som KOMBIT anvender, samt at den understøtter etableringen af en 2-vejs SSL forbindelse.

Kodebasen anvender XML-baseret konfiguration af WCF, og hvis man i sin egen kode ønsker at konfigurere WCF via kode og ikke XML, så er det vigtigt at man får alt fra App.config med fra eksempelkoden, da den indeholder konfiguration af OIO IDWS frameworket.

Klienten kører umiddelbart, såfremt man følger instruksen og installerer:

- .NETFramework 4.5.2.
- Microsoft Build Tools 2015 (Hvis du ikke ønsker at installere Visual Studio, men blot vil afvikle eksempelklienten)

og eksekverer run.bat fra en kommandoprompt. Gennemfør kald til demo-service

7.3 Step 2: Gennemfør test i ekstern test mod konkret webservice

Du har gennemført test mod demo-servicen og er nu klar til gå videre med implementeringen af Security Token Service. Nu kan du begynde din test op imod den konkrete service, du ønsker at benytte til din løsning.

7.3.1 Oprette en Serviceaftale på den konkrete service i ekstern test

Nu skal du oprette en serviceaftale på den service, du ønsker at teste forretningsmæssigt i forhold til din løsning.

- Log på STS Administration
- Opret en serviceaftale på den konkrete service du ønsker at anvende
- Få kommunen eller helpdesk* til at godkende serviceaftalen
- Noter følgende parameter fra Serviceaftalen
    - CVR-nr på myndigheden
    - Service EntityId
    - Service Endpoint

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen. Du kan læse mere om hvordan du opretter serviceaftaler i vejledningen til STS Administration [ADMINISTRATION].

7.3.2 Installer og udskift de nødvendige certifikater

Filen ConfigVariables.cs, som du finder sammen demo-servicen, indeholder en lang stribe konfigurationsparametre, og den er pre-konfigureret til at kalde en test-service på Serviceplatformens testmiljø, for et specifikt CVR nummer.

Derfor skal du nu tilpasse kodebasen, så du kan teste din løsning op i mod den service du ønsker at kalde. Det betyder skal du skifte certifikatet ud med dit eget certifikat – det som du har registreret på dit it-system i STS Administration. Dernæst skifter pegepinden ud så den peger på den service som du ønsker at kalde.

Du finde en kort vejledning til at udskifte certifikat i roden af kodebasen, og selve WSDL/XSD filer til den service man kalder skal skiftes i denne folder

https://github.com/Serviceplatformen/demoservice-client-java/tree/master/demoservice-token-client/src/main/resources/contracts

Husk at disse parametre skal du ændre, så de stemmer overens med henholdsvis miljø, certifikater, service og myndighed, som du ønsker at gøre brug af.

Der er endvidere oprettet en Service Reference til test-servicen på Serviceplatformen, og der skal oprettes en ny Service Reference til den faktiske service man ønsker at kalde. Når man opretter en ny Service Reference, er det vigtigt at man får koblet Bindings og Behaviours (se App.config) på denne service, så den rigtige sikkerhedsmodel anvendes i kaldet til servicen.

Når du foretaget ovenstående ændringer skal du trække et token til den valgte service og validere at du får udstedt et token til servicen på vegne af en kommune

7.3.3 Anbefalede testcases

- Genbrug af token:

  Tokens har en levetid på 8 timer, hvilket betyder at du skal genbruge tokenet.

- Hold live i flere tokens med forskellige myndigheder:
  Mange løsninger har flere kommuner på, og derfor er det vigtigt at din løsning kan håndtere at holde live i flere forskellige tokens og genbruge dem på de rigtige myndigheder.

-

7.3.4 Test af forretningslogik på konkret service

Når du har styr på at trække tokens kan du afvikle dine planlagte forretningstest af den valgte service.

Du kan finde inspiration til test af de fleste services i testproduktkataloget og i servicekataloget hvor du finder en links til relevant dokumenataion i forhodl til den enkelte service

7.4 Fase 3: Ibrugtagning i produktion

7.4.1 Registrer it-system i produktion

Før du kan gennemføre kald i produktion skal du registre dit it-system i produktion.

Du skal sørge for, at

- Oprette din organisation i STS Administration i produktion
- Registrere dit it-system i STS administration
- Tilknytte et gyldigt FOCES certifikat med SAML metadata til dit it-system

Hvis din organisation endnu ikke er oprettet på produktionmiljøet skal du kontakte helpdesk@serviceplatformen.dk

7.4.2 Opret en Serviceaftale på konkret service i produktion

- Log på STS Administration
- Opret en serviceaftale på demo-service
- Få kommunen* til at godkende serviceaftalen
- Noter følgende parameter fra Serviceaftalen
    - CVR-nr på myndighed
    - Service EntityId
    - Service Endpoint

Du finder paramenterne ved at klikke på informationsikonet i serviceaftalen.

8 KODEEKSEMPLER

8.1 Kodeeksempler til Invocation Context

Her kan du se et eksempel på hvordan værdierne indsættes i XML'en

CPR Kommunekode Service – InvocationContext

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://serviceplatformen.dk/xml/wsdl/soap11/CPRMunicipalityCodeService/1/"
xmlns:invctx="http://serviceplatformen.dk/xml/schemas/InvocationContext/1/"
xmlns:ns2="http://serviceplatformen.dk/xml/schemas/cpr/PNR/1/">
<soapenv:Header/>
<soapenv:Body>
<ns:getMunicipalityCodeRequest>
<invctx:InvocationContext>
              <invctx:ServiceAgreementUUID>ServiceAgreementUUID</invctx:ServiceAgreementUUID>
              <invctx:UserSystemUUID>UserSystemUUID</invctx:UserSystemUUID>
              <invctx:UserUUID>UserUUID</invctx:UserUUID>
              <invctx:ServiceUUID>cdf2d790-63ed-11e3-96c7-000c7631174f</invctx:ServiceUUID>
</invctx:InvocationContext>
<ns2:PNR>0000000000</ns2:PNR>
</ns:getMunicipalityCodeRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Her har du et eksempel hvor de valgfrie felter også er udfyldt.
Det er et eksempel på "CVR service":

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:x.="http://rep.oio.dk/eogs/xml.wsdl/"
xmlns:ns="http://serviceplatformen.dk/xml/schemas/InvocationContext/1/"
xmlns:x.1="http://rep.oio.dk/eogs/xml.schema/">
<soapenv:Header/>
<soapenv:Body> 4
<x.:GetLegalUnitRequest level="2">
<ns:InvocationContext>
<ns:ServiceAgreementUUID>43fb7e80-3f80-11e2-a32b-d4bed98c63db</ns:ServiceAgreementUUID>
<ns:UserSystemUUID>17b22dc2-3f80-11e2-a32b-d4bed98c63db</ns:UserSystemUUID>
<ns:UserUUID>fb21b665-3f7f-11e2-a32b-d4bed98c63db</ns:UserUUID>
<ns:ServiceUUID>93a48b42-3945-11e2-9724-d4bed98c63db</ns:ServiceUUID>
<ns:CallersServiceCallIdentifier>DEMO</ns:CallersServiceCallIdentifier>
</ns:InvocationContext>
<x.1:LegalUnitIdentifier>78834412</x.1:LegalUnitIdentifier>
</x.:GetLegalUnitRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Redirect services kaldes på same made som andre services bortset fra at kaldet inkluderer invocationContext parametre som URL parametre
An example of a redirect call to the "Post Danmark FindOs Service":
https://prod.serviceplatformen.dk/service/Redirect/Redirect/1 ?serviceAgreementUUID=*uuid* &userSystemUUID=*uuid* &userUUID=*uuid* &onBehalfOfUser=kbh &serviceUUID=*uuid* &callersServiceCallIdentifier=kombit &accountingInfo=account &postnr=4000 &husnr=1 &vejnavn=Algade where the *uuid* parts are meant to be proper UUIDs[3].
[3] UUIDs are in the canonical form.
The client sending the request must be able to follow redirects (HTTP Code 307), and must use the same certificate in the call to the Serviceplatform as it would use in a call to the service directly.

8.2 Kodeeksempel til Authority Context

Nedenfor ses et eksempel fra .NET demo-klienten:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
   <s:Header>
```

```
      <ActivityId CorrelationId="8259f8a7-e741-4ac7-a6da-e8a660609b10"
      xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">00000000-0000-0000-0000-
      000000000000</ActivityId>
   </s:Header>
   <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <CallDemoServiceRequest xmlns="http://serviceplatformen.dk/xml/wsdl/soap11/SP/Demo/1/">
         <CallContext xmlns="http://serviceplatformen.dk/xml/schemas/CallContext/1/">
            <OnBehalfOfUser>behalfOfUserDemoServicePortType.test.uuid</OnBehalfOfUser>
            <CallersServiceCallIdentifier>.netDemoClientCall</CallersServiceCallIdentifier>
            <AccountingInfo>ed8aea2d-a6a4-40a1-bf5e-6adf583254ff</AccountingInfo>
         </CallContext>
         <AuthorityContext xmlns="http://serviceplatformen.dk/xml/schemas/AuthorityContext/1/">
            <MunicipalityCVR>29189846</MunicipalityCVR>
         </AuthorityContext>
         <messageString>Hello Jesper!%&</messageString>
      </CallDemoServiceRequest>
   </s:Body>
</s:Envelope>
```

Som det kan ses ovenfor er både AuthorityContext og CallContext udfyldt.

8.3 Kodeeksempel til Security Token Service

```
<wsp:Policy wsu:Id="TokenAsymmetricBindingPolicy">
    <wsp:All>
        <wsam:Addressing wsp:Optional="false">
            <wsp:Policy/>
        </wsam:Addressing>
        <sp:SignedParts>
            <sp:Body/>
            <sp:Header Name="MessageID" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="RelatesTo" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="Action" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="To" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="From" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="ReplyTo" Namespace="http://www.w3.org/2005/08/addressing"/>
            <sp:Header Name="Framework" Namespace="urn:liberty:sb:2006-08"/>
        </sp:SignedParts>
        <sp:AsymmetricBinding>
            <wsp:Policy>
                <sp:InitiatorToken>
                    <wsp:Policy>
                        <sp:SamlToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/Never">
                            <wsp:Policy>
                                <sp:WssSamlV20Token11/>
                            </wsp:Policy>
                        </sp:SamlToken>
                    </wsp:Policy>
                </sp:InitiatorToken>
                <sp:RecipientToken>
                    <wsp:Policy>
                        <sp:X509Token sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToInitiator">
                            <wsp:Policy>
                                <sp:WssX509V3Token10/>
                            </wsp:Policy>
                        </sp:X509Token>
                    </wsp:Policy>
                </sp:RecipientToken>
                <sp:AlgorithmSuite>
                    <wsp:Policy>
                        <sp:Basic256Sha256/>
                    </wsp:Policy>
                </sp:AlgorithmSuite>
                <sp:Layout>
                    <wsp:Policy>
                        <sp:Strict/>
                    </wsp:Policy>
                </sp:Layout>
                <sp:ProtectTokens/>
                <sp:IncludeTimestamp/>
                <sp:OnlySignEntireHeadersAndBody/>
            </wsp:Policy>
        </sp:AsymmetricBinding>
        <sp:SignedSupportingTokens>
            <wsp:Policy>
                <sp:IssuedToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
                    <sp:RequestSecurityTokenTemplate/>
                    <wsp:Policy>
                        <sp:WssSamlV20Token11/>
                    </wsp:Policy>
                </sp:IssuedToken>
            </wsp:Policy>
        </sp:SignedSupportingTokens>
    </wsp:All>
</wsp:Policy>
```

8.4 Typiske fejl i forbindelse med implementering af Security Token

Nedenfor er der en liste af de oftest forekommende fejl, og løsning til disse:

Fejl	Løsning
Ikke-afviklet undtagelse: System.ServiceModel.Security.MessageSecurityException: Id-kontrollen lykkedes ikke for en udgående meddelelse. Det forventede DNS-id for fjernslutpunktet var SP_TEST_JJN_EBOKS (funktionscertifikat), men fjernslutpunktet angav DNS-kravet kombit-sp-signing-test (funktionscertifikat). Hvis dette er et legitimt fjernslutpunkt, kan du rette problemet ved eksplicit at angive DNS-id'et kombit-sp-signing-test (funktionscertifikat) som egenskaben Identity for EndpointAddress, når kanalproxien oprettes.	Forkert offentlig certifikat til verifikation af signerede beskeder fra Serviceplatformen Ret i fil ConfigVariables.cs til følgende : public const string ServiceCertificateAlias = "kombit-sp-signing-test (funktionscertifikat)";
Ikke-afviklet undtagelse: System.ServiceModel.FaultException: The received usercontext doesn't exist on the anvendersystem	Forkert Cvr-nr er anvendt. Ret i fil configVariables.cs til korrekt CVR-nr: public const string Cvr = "xxxxxxxx";
Ikke-afviklet undtagelse: System.ServiceModel.FaultException: Path resolution: no connection 'http://demo.prod-serviceplatformen.dk/service/DemoService/2' found	Entityid for service er forkert. Ret i fil ConfigVariables.cs til korrekt EntityID : public const string ServiceEntityId = ……”
Ikke-afviklet undtagelse: System.ServiceModel.Security.MessageSecurityException: Id-kontrollen lykkedes ikke for en udgående meddelelse. Det forventede DNS-id for fjernslutpunktet var SP_TEST_JJN_EBOKS (funktionscertifikat), men fjernslutpunktet angav DNS-kravet KOMBIT Støttesystemer-T (funktionscertifikat). Hvis dette er et legitimt fjernslutpunkt, kan du rette problemet ved eksplicit at angive DNS-id'et KOMBIT Støttesystemer-T (funktionscertifikat) som egenskaben Identity for EndpointAddress, når kanalproxien oprettes.	Forkert angivelse af STS certifikate. Ret i fil ConfigVariables.cs til: public const string StsCertificateAlias = "KOMBIT Støttesystemer-T (funktionscertifikat)";
Ikke-afviklet undtagelse: System.ServiceModel.Security.MessageSecurityException: Id-kontrollen lykkedes ikke for en udgående meddelelse. Det forventede DNS-id for fjernslutpunktet var KOMBIT Støttesystemer-T (funktionscertifikat), men fjernslutpunktet angav DNS-kravet SP_TEST_JJN_EBOKS (funktionscertifikat). Hvis dette er et legitimt fjernslutpunkt, kan du rette problemet ved eksplicit at angive DNS-id'et SP_TEST_JJN_EBOKS (funktionscertifikat) som egenskaben Identity for EndpointAddress, når kanalproxien oprettes.	Forkert angivelse af STS certifikat aftryk. Ret i fil ConfigVariables.cs til: //public const string StsCertificateThumbprint = "3a 10 3c f6 0a 9d 97 7b a5 a4 99 3e 06 d8 c9 ba 36 f2 58 3e";

Ikke-afviklet undtagelse: System.ArgumentException: No certificate with thumbprint 19 0D 33 DF EA 1B FB EE EE 18 63 09 42 4B C3 5C CB F9 07 36 is found.	Forkert angivelse af anvender funktionscertifikat aftryk. Check aftryk for funktions certifikat og ret i fil ConfigVariables.cs til korrekt aftryk. public const string ClientCertificateThumbprint = ””
Ikke-afviklet undtagelse: System.ServiceModel.EndpointNotFoundException: Der var ikke noget slutpunkt, der lytter på https://exttest.serviceplatformen.dk/service/SP/Demo/2/, og som kan acceptere meddelelsen. Dette skyldes ofte en forkert adresse eller SOAP-handling. Du kan finde flere oplysninger under InnerException. ---> System.Net.WebException: Fjernserveren returnerede en fejl: (404) Ikke fundet. ved System.Net.HttpWebRequest.GetResponse()	Forkert angivelse af endpoint. Ret I DemoTokenClient.exe.config til korrekt end point. <endpoint address="https://exttest.serviceplatformen.dk:443/service/SP/Demo/1/" behaviorConfiguration="LibBasBehaviourConfiguration" binding="LibBasBinding" bindingConfiguration="LibBasBindingConfiguration" contract="DemoService.DemoPortType" name="DemoTokenClient" />

9 FEJLBESKEDER

I nedenstående findes de typiske fejlbeskeder og løsninger.

9.1 ServiceUUID er forkert i forhold til den kaldte service

Hvis Serviceplatformen returnerer en fejl kontrollér da den modtagne fejlbesked. Der gives typisk en konkret fejlbeskrivelse hvis f.eks. ServiceUUID er forkert i forhold til den kaldte service.

Dette giver fejlen:

```
com.systematic.kombit.common.exception.MessageKombitException: Service called with wrong service
UUID '<Kaldt ServiceUUID>'
```

9.2 Forkert syntaks for det angivne ServiceUUID

Dette kan skyldes at der er tastet et mellemrum eller tabulering foran eller bagved UUID'et. Dette giver fejlen:

```
com.systematic.kombit.common.exception.MessageKombitException: Provided service UUID '<Kaldt
ServiceUUID>' is not a UUID.
```

9.3 KOMBIT token er udløbet

Tokens udstedt af rammearkitekturens STS er kun gyldig i en begrænset tidsperiode. Sendes en udløbet token til Serviceplatformen, vil dette give følgende fejl:

```
com.systematic.kombit.common.exception.MessageKombitException:  Security token is expired: <copy of
token content>
```

9.4 Forskelligt certifikat brugt til webservice kald og STS token udstedning

Certifikatet benyttet til oprettelse af aftale i STS skal også benyttes til oprettelse af aftlale på Serviceplatformen og ved kald foretaget til Serviceplatformen. Stemmer disse ikke overens, vil dette give følgende fejl:

```
com.systematic.kombit.common.exception.MessageKombitException: Client certificate did not match the
subject certificate of the security token
```

9.5 Serviceaftale fra støttesystemet er udløbet

Ved kald med KOMBIT token eller AuthorityContext fremsøges den adgangsgivende aftale som udgangspunkt i støttesystemets sikkerhed. Såfremt aftalen er udløbet (eller endnu ikke er gyldig) vil dette give følgende fejl:

```
com.systematic.kombit.common.exception.ConfigurationKombitException: No valid service agreement
could be found for the given municipality and certificate.
Contact RA-STS support for help. The UUIDs of the invalid alternatives were:
  <Benyttet ServiceAgreementUUID> (outside validity period <Gyldighedsperiode>)
```

Sker ovenstående fejl, kontakt da supporten for rammearkitekturens støttesystemer (RA-STS) og *ikke* Serviceplatformens Helpdesk.

Vær opmærksom på at fejlbeskeden kan indeholde mere end én ugyldig serviceaftale.

9.6 Serviceaftale fra støttesystemet er (midlertidigt) opsagt

Ved kald med KOMBIT token eller AuthorityContext fremsøges den adgangsgivende aftale som udgangspunkt i støttesystemets sikkerhed. Såfremt aftalen er opsagt vil dette give følgende fejl:

```
com.systematic.kombit.common.exception.ConfigurationKombitException: No valid service agreement
could be found for the given municipality and certificate.
Contact RA-STS support for help. The UUIDs of the invalid alternatives were:
  <Benyttet ServiceAgreementUUID> (suspended for service <Kaldt services UUID>)
```

Sker ovenstående fejl, kontakt da supporten for rammearkitekturens støttesystemer (RA-STS) og *ikke* Serviceplatformens Helpdesk.

Vær opmærksom på at fejlbeskeden kan indeholde mere end én ugyldig serviceaftale.

10 APPENDIX A – OPSÆTNING AF SOAPUI TIL INVOCATION OG AUTORITY CONTEXT

10.1 opsætning af SoapUI klient

En god start til at teste kald mod Serviceplatformen, med invocation og authority -context, er at opsætte SoapUI.

Via SoapUI, kan der hurtigt testes, om eksempelvis udfyldelse af InvocationContext er korrekt og hvordan strukturen af svaret ser ud.

10.1.1 Opsætning af keystore

SoapUI skal bruge et funktionscertifikat pakket ind i Javas keystore-format for at kunne læse indholdet. Dette oprettes ved at læse virksomheds- (VOCES) / funktionscertifikatet (FOCES) ud i en JKS fil via Java- værktøjet keytool.

Dette køres i en kommandolinje (i windows: cmd.exe) Kaldet, der skal udføres, ser således ud:

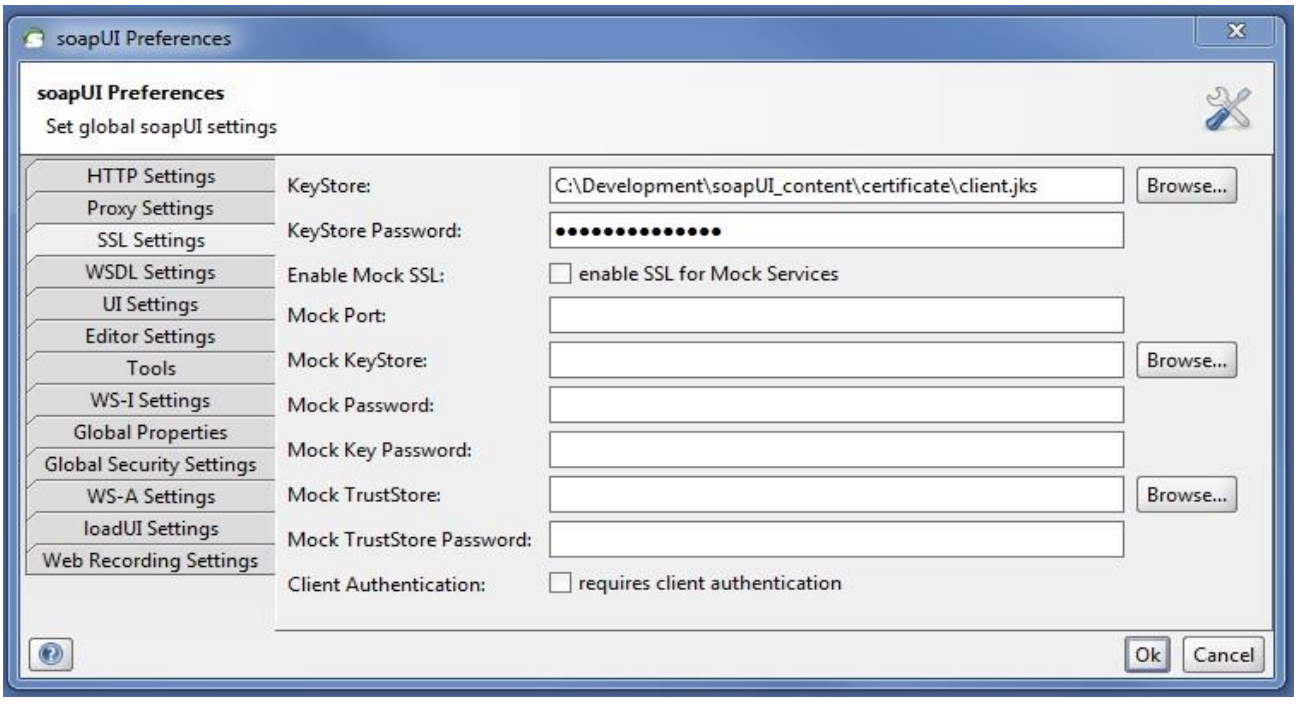
```
keytool -v -importkeystore -srckeystore "<filnavn>.p12" -srcstoretype PKCS12 -destkeystore funkcert.jks
```

Den fil der dannes, skal beskyttes af et password, som ikke behøver at være det samme password som til certifikatet. Nogle værktøjer (fx Visual Studio) kræver dog, at de to passwords er ens.

Herefter åbnes SoapUI på følgende måde (Figur 5):

1. Vælg File → Preferences
2. Vælg fanebladet SSL Settings og udfyld stien til KeyStore og KeyStore Password
3. Tryk ok.

VOCES/FOCES certifikat er nu klar til at blive brugt med kaldet mod Serviceplatformen.



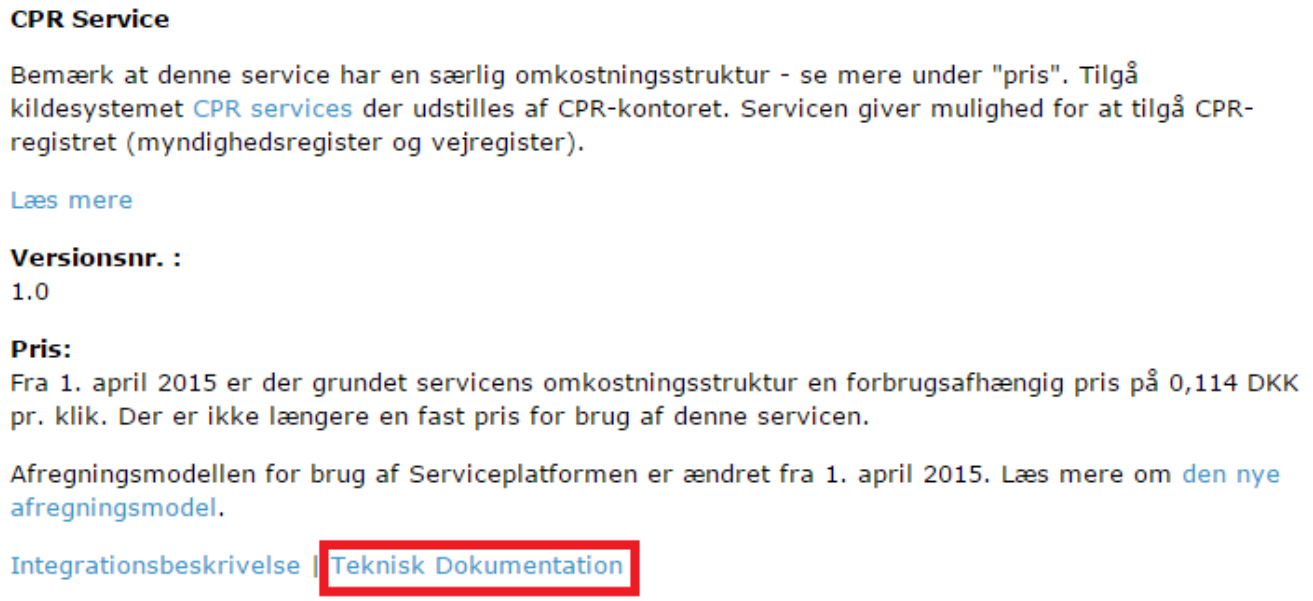
Figur 5 – Viser hvilket certifikat der bruges af SoapUI

10.1.2 **Hent WSDL fil**

En WSDL-fil beskriver hvilken struktur, webservicen forventer at modtage fra klienten, og hvilken struktur webservicen returnerer. Denne WSDL-fil bruges af SoapUI til at opbygge XML strukturen af webservicekaldet.

WSDL-filer findes for hver version af en service under menupunktet ”Find service”. Vælg den service, du ønsker at kalde, og klik på ”Teknisk Dokumentation” (se Figur 6) for at downloade et ZIP arkiv som bl.a. indeholder servicens WSDL-fil.

I dette eksempel bruges CPR Service version 1.0 og dennes WSDL-fil. Den benyttede sikkerhedsmodel i eksemplet er InvocationContext.



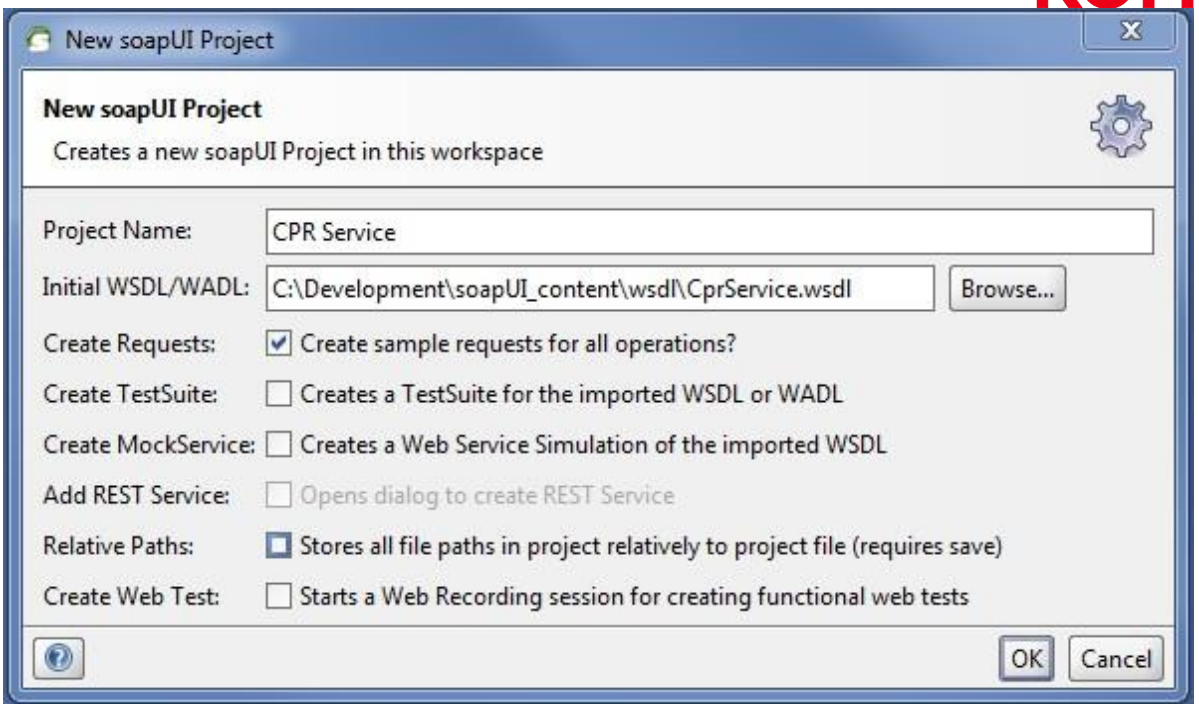
Figur 6 – Viser hvor bruger kan finde Service WSDL’en

10.1.3 **Opret projekt**

Dernæst opret et nyt projekt i SoapUI (Figur 7), for at opsætte kaldet mod en given webservice på Serviceplatformen.

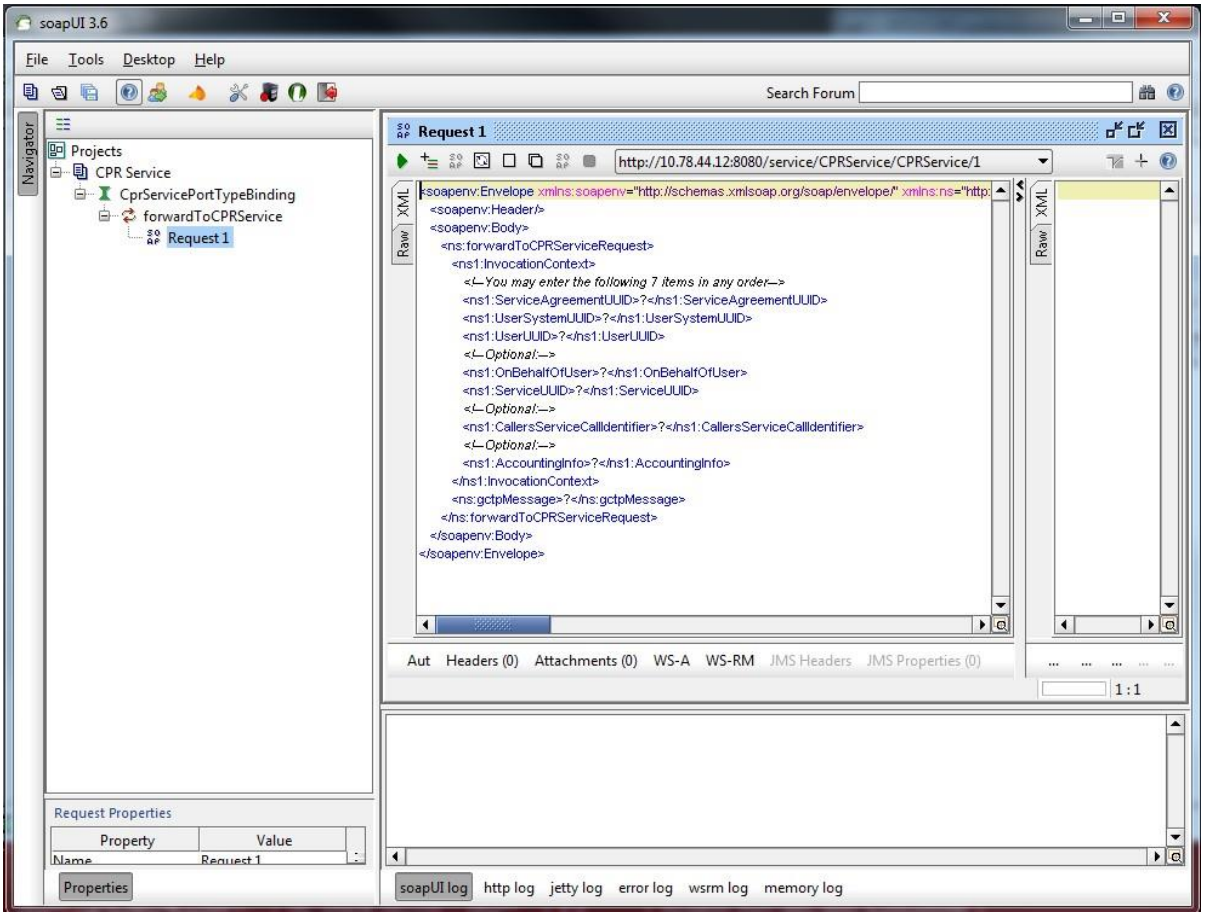
Projektet oprettes således:

1. Angiv et projektnavn.
2. Indsæt stien til den downloadede og udpakkede WSDL-fil.
3. Klik på OK



Figur 7 – Oprettelse af SoapUI projekt

SoapUI opretter et projekt med et tomt request/kald ud fra WSDL-filen. Klik ind på projektet og kontrollér, at alt er oprettet korrekt for callGctpService som figuren nedenfor (Figur 8). Dette vil vise, at SoapUI har oprettet en XML struktur, som nu kan modificeres med bl.a. værdier i InvocationContext.

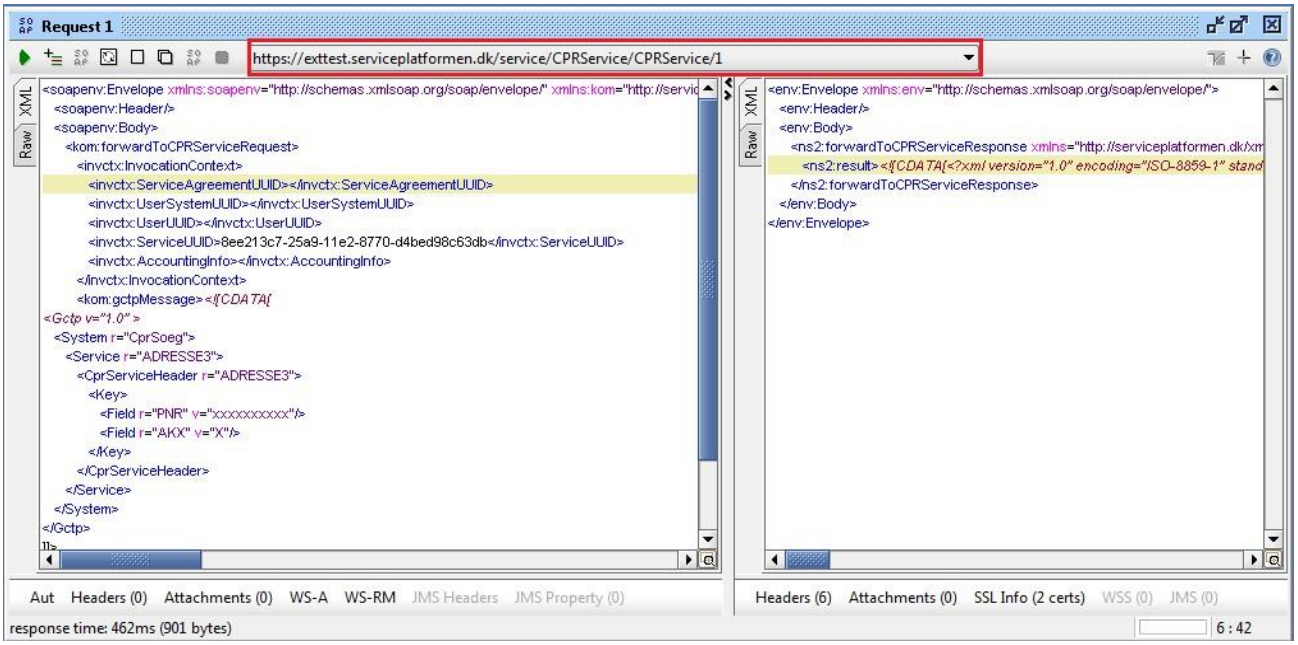


Figur 8 – Oprettet xml struktur fra WSDL

10.1.3.1 Ændre endpoint

I WSDL-filen er endpointet (stien til kald af servicen) ikke sat korrekt, så først skal man klikke på dropdown-pilen (Figur 9) ud for endpointet (der starter med http://10.78...) og vælge [add new endpoint…]. Indsæt et nyt endpoint med adressen:

https://exttest.serviceplatformen.dk/service/CPRService/CPRService/1

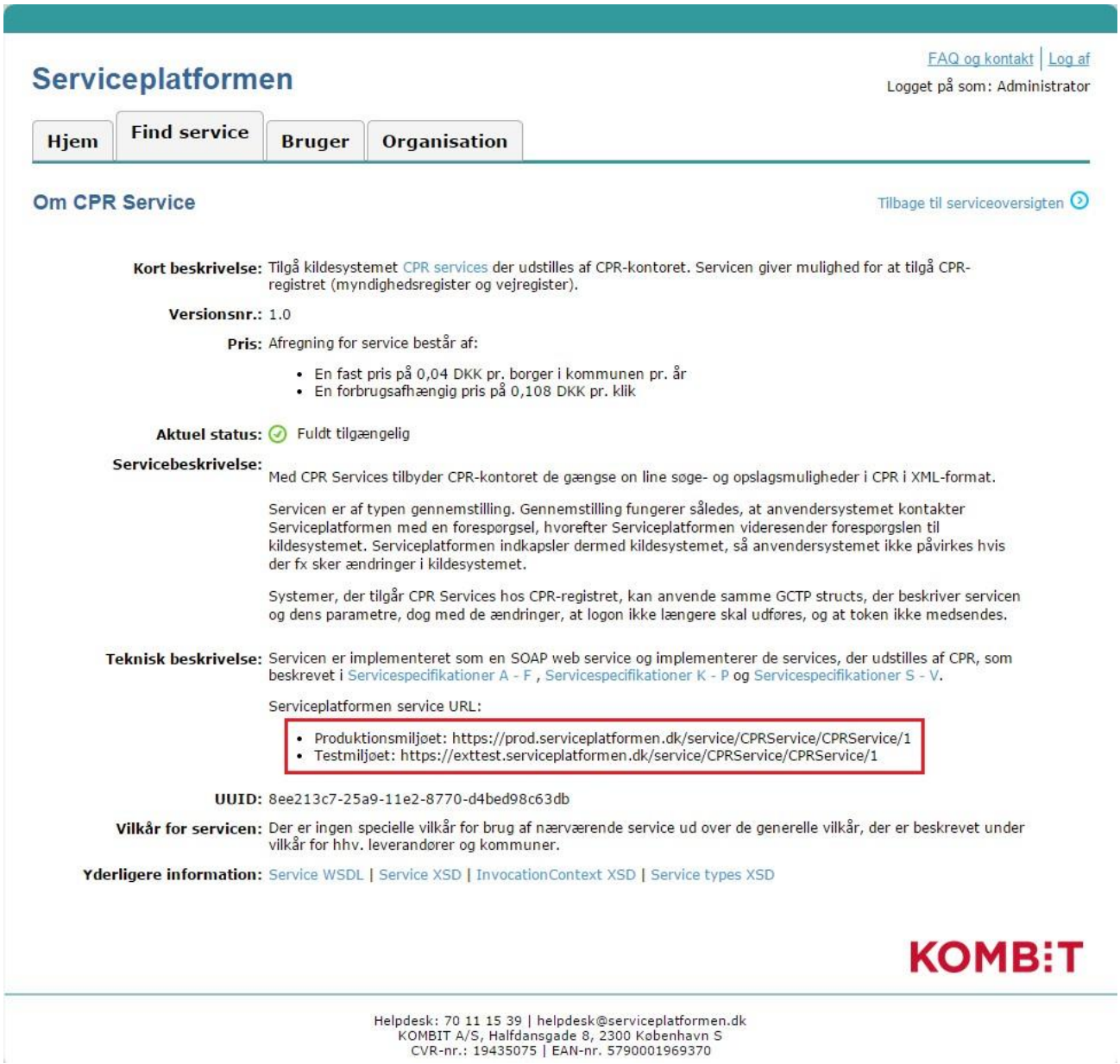


Figur 9 – Ændring af endpoint / URL til servicekald

Alle endpoints kan findes under ”Find service” på konkrete service, ved klik på ”Læs mere” (Figur 10).

Informationerne kan tilgåes fra følgende links:

Miljø	Link
Ekstern test	https://exttestwww.serviceplatformen.dk/administration/serviceOverview/list
Produktion	https://www.serviceplatformen.dk/administration/serviceOverview/list



Figur 10

10.1.3.2 Opsætning af SOAP og GCTP kald

XML kaldet skal opdateres, så alle oplysningerne passer (Se venligst nedenstående XML). Dette betyder at værdier i InvocationContext skal indsættes med de UUID'er som findes for jeres respektive serviceaftaler

Udfyld InvocationContext. Værdierne, der skal indsættes i InvocationContext, er beskrevet i afsnittet Udfyldelse af InvocationContext.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:kom="http://serviceplatformen.dk/xml/wsdl/soap11/CprService/1/"
xmlns:invctx="http://serviceplatformen.dk/xml/schemas/InvocationContext/1/">
```

```
<soapenv:Header/>
<soapenv:Body>
<kom:forwardToCPRServiceRequest>
<invctx:InvocationContext>
              <invctx:ServiceAgreementUUID>ServiceAgreementUUID</invctx:ServiceAgreementUUID>
              <invctx:UserSystemUUID>UserSystemUUID </invctx:UserSystemUUID>
              <invctx:UserUUID>UserUUID </invctx:UserUUID>
              <invctx:ServiceUUID>8ee213c7-25a9-11e2-8770-d4bed98c63db</invctx:ServiceUUID>
</invctx:InvocationContext>
<kom:gctpMessage>
              <![CDATA[
<Gctp v="1.0" >
<System r="CprSoeg">
<Service r="ADRESSE3">
<CprServiceHeader r="ADRESSE3">
<Key>
              <Field r="PNR" v="xxxxxxxxxx"/>
              <Field r="AKX" v="X"/>
</Key>
</CprServiceHeader>
</Service>
</System>
</Gctp>
]]>
</kom:gctpMessage>
</kom:forwardToCPRServiceRequest>
</soapenv:Body>
</soapenv:Envelope>
```

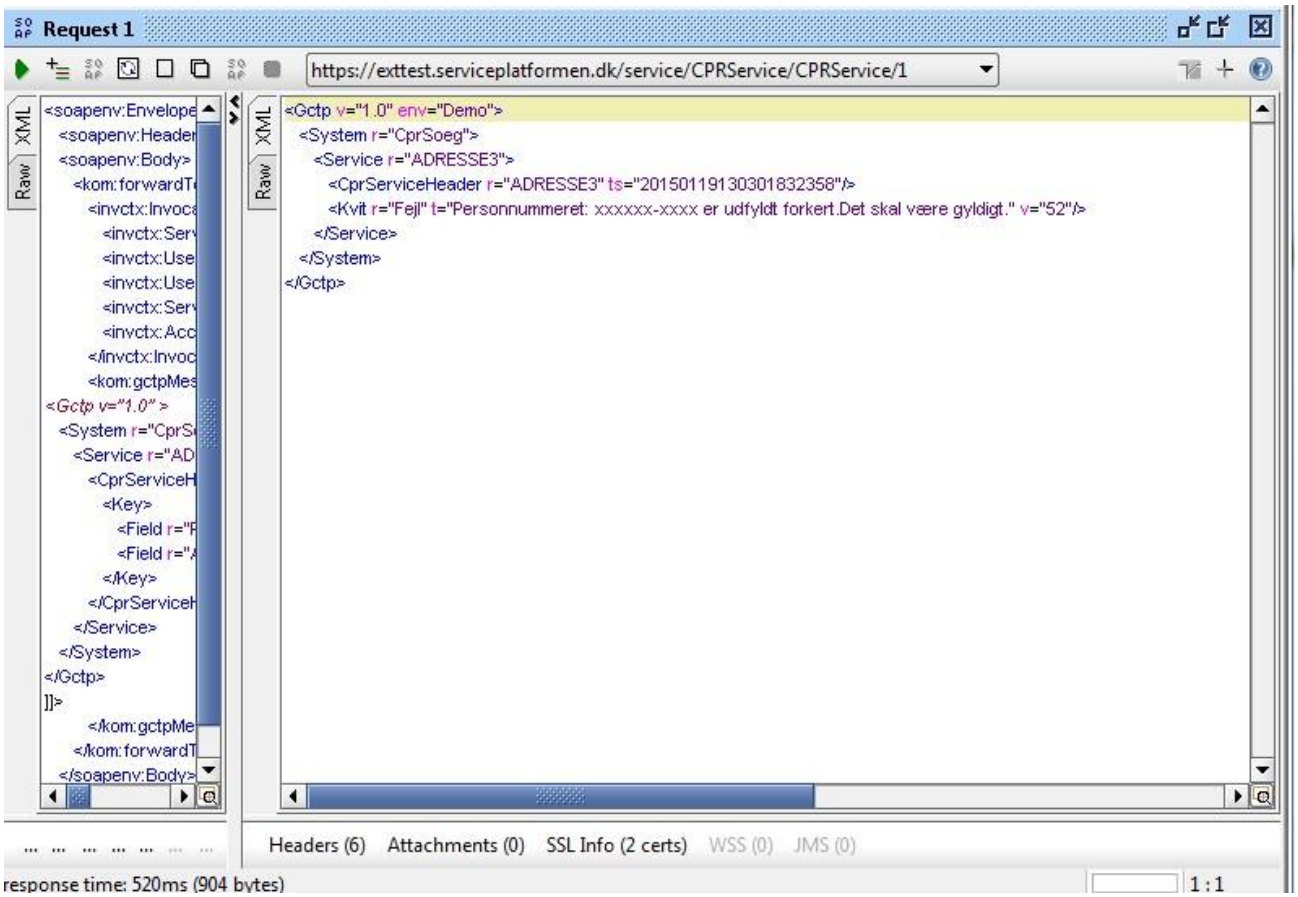
10.1.3.3 1.5.1.6 Udførsel af SOAP og GCTP kald

Når XML'en er sat op, trykkes på "Play"-knappen og kaldet udføres. I det returnerede svar, er også CDATA-indpakket og uden linjeskift.

Eksempel:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Header/>
<env:Body>
<ns2:forwardToCPRServiceResponse
xmlns="http://serviceplatformen.dk/xml/schemas/InvocationContext/1/"
xmlns:ns2="http://serviceplatformen.dk/xml/wsdl/soap11/CprService/1/">
<ns2:result><![CDATA[... GCTP-XML ......]]></ns2:result>
</ns2:forwardToCPRServiceResponse>
</env:Body>
</env:Envelope>
```

For nemmere at læse svaret fra eksempelvis CPR, kan man vælge at fjerne alt andet end GCTP-XML'en fra tekstboksen, og derefter trykke på Alt+F for at vise XML'en pænt formateret med linieskift (Figur 11).



Figur 11 – Viser formateret xml, med linieskift

11 REFERENCELISTE

Nedenfor finder du links til de bilag, vejledningen refererer til. Links fører dig til listevisninger i vores dokumentbibliotek. Listerne kan indeholde flere dokumenter- vær derfor særlig opmærksom på dokumentets titel, så du får fat i det rigtige dokument.

Ref.	Titel	Kommentarer
[ADMINISTRATION]	Brugervejledning til Administrationsmodulet for leverandører	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter
[META DATA]	Registrering af Metadata til KOMBITs testmiljø og produktionsmiljø	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter
[DANID]		https://www.nets-danid.dk/
[DEMO]		https://www.serviceplatformen.dk/administration/help/provider-tech-guide
[SAML]		https://www.oasis-open.org/standards#samlv2.0
[OIOSAML]		https://www.digitaliser.dk/group/42063
[ROLLEDESIGN]	Jobfuntionsroller - principper	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter
[KOMMUNEUUID]	KommuneUUID til InvocationContext	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter

[NEMLOGIN]		https://digst.dk/it-loesninger/nemlog-in/
[CFX]		https://www.digitaliser.dk/resource/3949909
[OIO IDWS 3.0]		https://www.digitaliser.dk/resource/3949880
[STS-VILKÅR]	"Bilag 2 - Vilkår for anvendelse af sikkerhedsmodellen i Rammearkitekturen version 2.2"	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter
[SIKKERHEDSMODEL]	Bilag 2A - Beskrivelse af sikkerhedsmodellen i Rammearkitekturen v.2.2	Digitaliseringskatalogets informationsside om den samlede adgangsstyringsløsning i den fælleskommunale rammearkitektur kan findes her: http://docs.kombit.dk/loesning/adgangsstyring/betingelse https://share-komm.kombit.dk/p089/Referencedokumenter