MedCom

Den Gode Webservice

En fælles webserviceprofil for sundhedsvæsenet

Version 1.0 - 13-07-2006

Introduktion

Den Gode Webservice (DGWS) har til formål at understøtte kommunikation med XML-webservices mellem de forskellige parter i sundhedssektoren – uafhængigt af, hvilke it-produkter og it-systemer de pågældende parter benytter.

Den Gode Webservice understøtter en **S**ervice **O**rienteret it-**A**rkitektur (SOA), som Ministeriet for Videnskab, Teknologi og Udvikling (MVTU) anbefaler, at den offentlige sektor anvender som fælles it-arkitektur.

Eksempel: Fremvisning af KPLL-laboratorieresultater på Sundhed.dk

Københavns Praktiserende Lægers Laboratorium (KPLL) har udviklet en webservice med navnet "sdnkpllWS". Webservicen gør det muligt for et klientsystem (Sundhed.dk) at hente KPLL-laboratorieresultater og fremvise dem på Sundhed.dk.

KPLLs webservice består af tre funktionskald: "GetPatientInfo", "GetPatientResults" og "GetAllPatientResults". Resultatet af forespørgslerne er en XML-fil, der indeholder et eller flere laboratoriesvar.

Selve laboratoriesvaret er et indsat XML-dokument, der svarer fuldstændig til et almindeligt laboratoriesvar, der overholder MedComs XML-standard "XRPT01" for laboratoriesvar.

Webservices gør det muligt at udveksle data på tværs af it-produkter og systemplatforme, men det sikrer ikke i sig selv, at produkterne og systemerne er interoperationelle.

Webservices kan anvendes på et utal af måder. Derfor er der behov for et fast defineret brugsmønster, en "profil", som præciserer, hvordan man skal anvende standarder for kommunikation og sikkerhed, og hvilket format fælles data skal have.

Den Gode Webservice definerer en sådan webserviceprofil for sundhedsvæsenet.

Profilen præciserer, hvordan sikkerhedsoplysninger udveksles i overensstemmelse med MVTUs retningslinjer for Offentlig Information Online (OiO) og for brug af digitale certifikater(OCES).

Dokumentet henvender sig til projektledere, løsningsdesignere, it-arkitekter og programmører.

Den Gode Webservice

- Er en SOAP-webservice profil, der fastlægger standarder for autentifikation og kommunikation af fælles sundhedsfaglige oplysninger mellem sundhedssektorens parter.

- Angiver, hvordan OCES-certifikater kan anvendes til autentifikation og digital signering i sundhedssektoren.

- Gør det muligt at kommunikere personhenførbare sundhedsoplysninger på en sikker og fleksibel måde.

- Gør det muligt for en webserviceudbyder at vælge mellem 5 forskellige sikkerhedsniveauer:
  - 5: Hele meddelelsen signeret med MOCES-signatur
  - 4: Id-kort signeret med MOCES-medarbejdersignatur
  - 3: Id-kort signeret med VOCES-virksomhedssignatur
  - 2: Username- og password-autentifikation
  - 1: Ingen personidentifikation.

- Angiver sikkerhedsprotokollen for simpel kommunikation mellem to parter og for en-til-mange- og mange-til-mange-kommunikation.

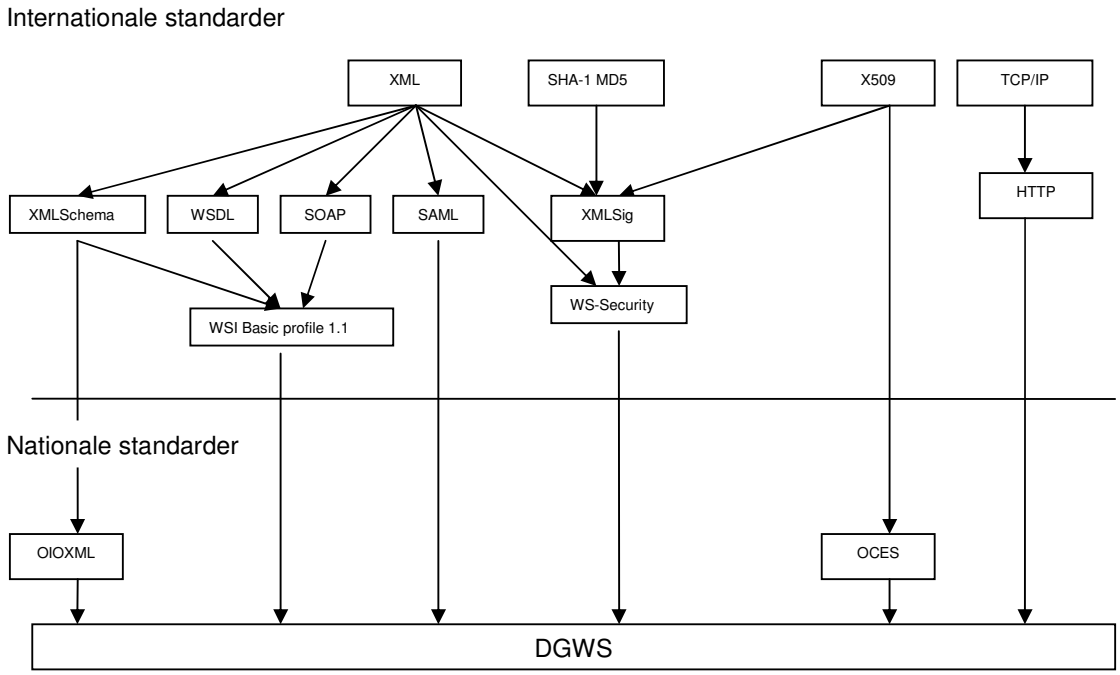
- Er uafhængig af den underliggende transportprotokol og kan f.eks. bruges med både VPN-kryptering på det lukkede sundhedsdatanet og med SSL-kryptering på det åbne internet.

- Beskriver, hvordan webservicekald grupperes som én arbejdsgang, og hvordan det afgøres, om arbejdsgangen er afsluttet og med hvilken status.

- Fastsætter retningslinjer for, hvordan services opbygges, så de bliver robuste over for timeouts og retransmissioner.

Anvendte standarder

Figuren nedenfor viser sammenhængen mellem de vigtigste standarder, der ligger til grund for Den Gode Webservice.



Nationale og internationale standarder og Den Gode Webservice.

Internationale standarder

XML Schema 1.1: XML-skemaer udtrykker en delt dataforståelse, der tillader maskinel behandling. Skemaerne definerer de konkrete XML-dokumenters struktur, indhold og semantik.

SOAP 1.1: SOAP er en XML-baseret generel "kuvertstandard" til webservicekommunikation. SOAP-kuverten består af to dele: en "header"-del, der indeholder autentifikations- og sikkerhedsdata, og en "body"-del, der indeholder selve kommunikationsindholdet.

HTTP 1.1: HyperText TransportProtokollen er fundamentet for webbaseret kommunikation, herunder gængs HTML. HTTP er en solid protokol, hvor kommunikationen "starter helt forfra" efter hver totrins-meddelelse – der hver består af en request og en efterfølgende response-meddelelse.

WS-Security 1.1: XML-baseret sikkerhedsspecifikation. Anvendes til kommunikation af autentifikationsdata, digitale signaturer mv.

RSA: Krypteringsalgoritme for "public key encryption". Her anvendes der et nøglepar, som består af en offentlig nøgle og en privat nøgle. Den offentlige nøgle er kendt af alle, og den private kun af den enkelte bruger. RSA anvendes i OCES-certifikater og benyttes bl.a. til at signere data.

SHA-1: Mekanisme til at beregne en hash-værdi (et fingeraftryk) af et dokument. Hash-værdien har den egenskab, at to forskellige dokumenter i praksis aldrig giver samme hash-værdi. Hash-værdien underskrives med en nøgle for at danne en digital signatur af et dokument.

XMLSig: Sikrer oprindelse og integritet af XML-meddelelser og standardiserer den proces, hvorunder XML-indhold signeres og indsættes i XML-dokumenter.

SAML 2.0: Understøtter Single Signon ved at definere en ramme for udveksling af sikkerhedsoplysninger, herunder autentifikation og attributter, der kan anvendes til autorisation.

WSDL 1.1: Webservices Description Language er et XML-format til at beskrive servicesnitflader for webservices, dvs. datatyper, input, output, protokoller mv.

Nationale standarder

OIOXML: Dansk profil for, hvordan XML-skemaer udformes. Den er udarbejdet af Ministeriet for Videnskab, Teknologi og Udvikling (MVTU).

OCES: Dansk standard for digital signatur, der gør det muligt for brugerne at identificere sig i den digitale verden. Den indfrier de basale krav, som borgere, myndigheder og private virksomheder stiller til sikkerhed, når de skal udveksle fortrolige og følsomme oplysninger over internettet.

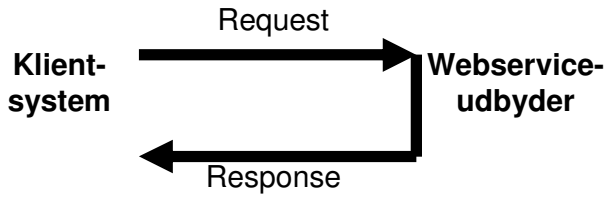
Grundlæggende arkitektur: Sådan anvendes Den Gode Webservice

SOA, HTTP og SOAP

Den Gode Webservice anvendes til at etablere system-til-system-integration med webservices. Integrationen bygger på principperne for den Service Orienterede Arkitektur (SOA).

SOA bygger på en filosofi om, at man skal uddelegere ansvaret og lade den, der er bedst til det, udføre arbejdet. I en SOA stiller it-systemer derfor services til rådighed, som andre it-systemer kan anvende til at løse en opgave. For at det kan lade sig gøre mellem mange parter, er det nødvendigt først at fastlægge fælles forventninger til bla. sikkerheden i snitfladen mellem udbydere og aftagere af services. Det er til gengæld ligegyldigt, hvilken teknisk platform de enkelte parter benytter, så længe de følger spillereglerne for udstilling og brug af services.

Den Gode Webservices kommunikationsmodel



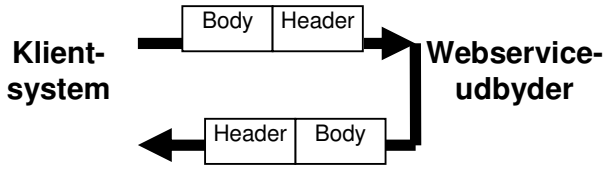
Den Gode Webservice bygger på internet-kommunikationsprotokollen ”HTTP”, der også anvendes ved kommunikation af hjemmesider. Kommunikationen sker i to-trins-meddelelser, der kaldes request (”kald”) og response (”svar”). HTTP-protokollen er en robust kommunikationsprotokol, hvor kommunikationen starter helt forfra efter hver request-response-sekvens.

SOAP-standarden er en standard for en webservicekuvert. Standarden opdeler request- og response-meddelelserne i en ”header”- og en ”body”-del.

- I **header** indsættes forsendelsesdata og sikkerhedsoplysninger. I Den Gode Webservice drejer det sig om tre typer data:

    1. **Forsendelsesoplysninger**, bl.a. meddelelsens nummer, prioritet og status.
    2. **Id-kort** med brugerens id og eventuelt andre brugeroplysninger, fx navn. Kortet kan i visse tilfælde være signeret, så modtageren har sikkerhed for, hvem der har afsendt meddelelsen, og for, at id-kortet ikke er ændret undervejs.
    3. **OCES Digital Signatur** bruges til at signere indholdet af hele SOAP-meddelelsen.

- I **body** indsættes de informationer, der vedrører den aktuelle webservice – altså selve "brevet" i SOAP-kuverten. Disse indlejrede XML-data beskrives ikke yderligere i denne dokumentation, men skal dokumenteres i beskrivelsen af de enkelte webservices.



I Den Gode Webservice anvender request- og response-meddelelserne samme XML-syntaks. For at sikre, at SOAP header og SOAP body altid benytter samme tegntabel, anvender Den Gode Webservice udelukkende UTF-8 Unicode standarden.

En webservice

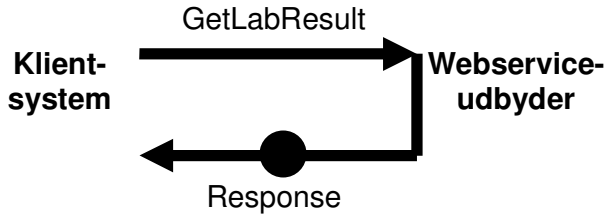
En webservice består af en række på forhånd fastlagte kald med tilhørende svar.

Webservicekommunikation involverer to parter: en webserviceudbyder og et klientsystem. Kommunikationen starter, når klienten kalder webservicen, og slutter, når klienten til sidst har modtaget svar på det sidste kald i den samlede webservice.

"Forespørgsel", "Meddelelse" og "Session" er eksempler på tre typer af webservices.

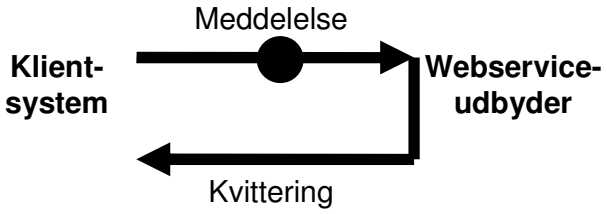
Simpel forespørgsel

Ved en simpel forespørgsel fremsender klienten en eller flere forespørgselsparametre i "request"-meddelelsen. I en laboratorieservice fremsendes f.eks. patientens CPR-nummer i forespørgslen. På baggrund af CPR-nummeret finder webservicen patientens laboratoriesvar, som den returnerer i body-delen af den efterfølgende "response"-meddelelse.



Meddelelse

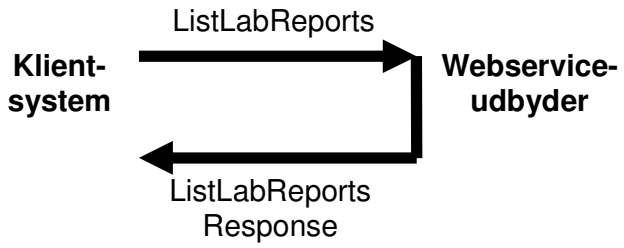
I en webservice kan body-delen af en request benyttes til at indsætte en separat XML-meddelelse, f.eks. en henvisning til et sygehus. Når webserviceudbyderen (her sygehuset) har modtaget laboratoriesvaret, returneres en positiv kvittering til klientsystemet i den efterfølgende response, og webservicen er afsluttet.



Session

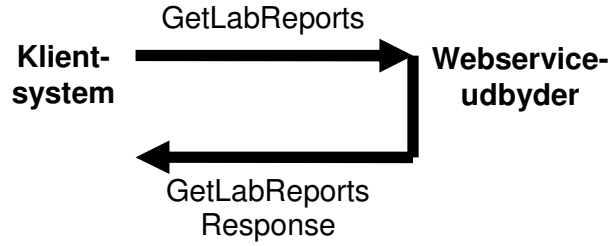
Mange webservices består af en række fastlagte webservice-kald og -svar. Det kaldes en session.

Eksempel



ListLabReports

Klientsystemet sender en patients CPR-nummer, og webservicen returnerer en liste med patientens laboratoriesvar.



GetLabReports

Klientsystemet fremsender et tidsinterval, og webservicen returnerer laboratoriesvarene fra den pågældende periode. Både forespørgslens CPR-nummer og laboratoriesvarene fremsendes som indsatte XML-meddelelser i SOAP-meddelelsernes body-del.

De enkelte kald i en webservice navngives med et navn, der beskriver kaldets funktion, f.eks. ”GetLabReports”. Det resulterende svar tilføjes ordet ”Response”, det vil her sige ”GetLabReportsResponse”. Som det fremgår senere, indgår disse servicenavne i WSDL-dokumentationen af webservicen.

Den Gode Webservice anvender HTTP som transportmekanisme mellem klientsystem og serviceudbyder. Desværre kan HTTP fejle, og det er udefineret, hvad der skal ske, hvis f.eks. klientsystemet aldrig får svar på en forespørgsel.

For at forebygge at dette sker, skal både klientsystem og webserviceudbyder forsyne hver request og response med et unikt id for den pågældende meddelelse og med et FlowID, der er unikt for hele den pågældende session.

- Alle afsendte meddelelser skal forsynes med et unikt MessageID, der aldrig må bruges til andre meddelelser igen af den pågældende afsender – heller ikke efter en geninstallation. Dog skal meddelelsen ved genfremsendelse bruge samme MessageID.
- Serviceudbyderen skal i første response i sessionen forsyne meddelelsen med et unikt FlowID (løbenummer) for den pågældende session.
- Klientsystem og webserviceudbyder skal genbruge FlowID’et i efterfølgende meddelelser i samme session.

For yderligere at sikre en robust kommunikation genfremsendes sikkerhedsoplysninger i alle kald i den pågældende session.

Ved hjælp af disse mekanismer er det muligt for et klientsystem:

- at genfremsende tidligere fremsendte requests uændret, hvis et kald skulle blive afbrudt
- at springe et kald over, hvis klientsystemet allerede har de informationer, der er nødvendige for at benytte senere eller andre kald i webservicen.

Det er således serviceudbyderens ansvar at sortere dubletter fra og returnere svaret igen, hvis en request med samme MessageID modtages.

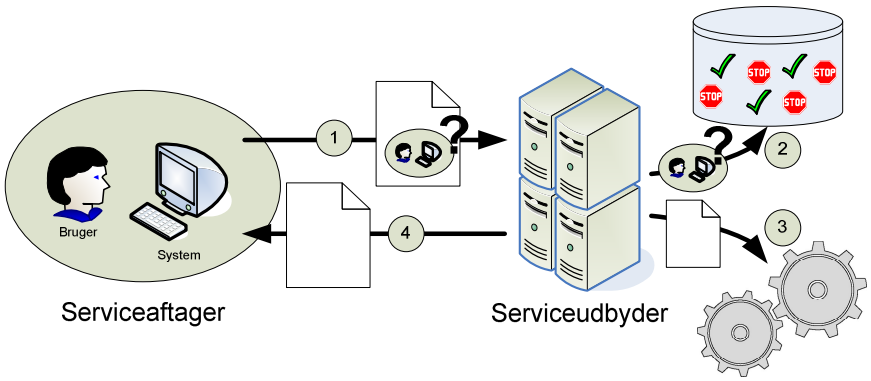
Webservice standarden WS-Secure Conversation definerer også en mekanisme til etablering af sessioner mellem to parter. DGWS benytter ikke denne specifikation, men forhindrer heller ikke at den i fremtiden kan blive anvendt om nødvendigt.

Arbejdsgangen i Den Gode Webservice

Al kommunikation med Den Gode Webservice involverer fire logiske operationer: autentifikation (identifikationen) af brugeren, autorisation (godkendelse) af brugeren, forespørgsel efter de ønskede data og returnering af svar eller fejl til klientsystemet.

Figuren nedenfor viser de fire operationer:

1. Klienten identificeres via de medsendte oplysninger om bruger og/eller system
2. Hvis autentifikationen lykkes, tjekkes klientens rettigheder i den lokale rettighedsdatabase for at se, om brugeren/systemet kan autoriseres til servicen.
3. Hvis brugeren/systemet kan godkendes, behandles forespørgslen.
4. Svaret returneres til klientsystemet



De fire logiske operationer i DGWS-arbejdsgangen

Sikkerhed

Datatilsynets persondatalov angiver reglerne for sikring af personfølsomme oplysninger, og patientretsstillingsloven (sundhedsloven) skærper kravene til sundhedspersoners behandling af fortrolige oplysninger.

Den Gode Webservice tager højde for de retslige krav om personsikkerhed ved at forholde sig til følgende sikkerhedsegenskaber:

- **Autentifikation**: Hvordan en bruger eller et system "logger" på en serviceudbyder og dermed identificerer sig selv.
- **Autorisation**: Hvordan en serviceudbyder kontrollerer, om en bruger eller et system har ret til at udføre en ønsket service.
- **Konfidentialitet**: Hvordan kuvertdata beskyttes mod uvedkommende under transporten mellem klient og webserviceudbyder.
- **Integritet**: Hvordan det sikres, at data ikke bliver modificeret under transporten mellem klient og webserviceudbyder.
- **Uafviselighed**: Hvordan det til enhver tid senere teknisk kan bevises, at en bruger eller et system har modtaget data fra en klient eller en webserviceudbyder.

Konfidentialiteten og integriteten forudsættes at blive sikret af transportlaget, f.eks. hvis det anvender SSL eller VPN. Kommunikeres der f.eks. via sundhedsdatanettet, er disse sikkerhedsaspekter garanteret, eftersom sundhedsdatanettet anvender VPN.

Ifølge sundhedssektorens nationale it-strategi for 2003-2007 er brugen af OCES-digitale certifikater helt central for sikkerheden i sundhedsvæsenet. Den Gode Webservice anbefaler derfor brugen af OCES-digitale certifikater som akkreditiver.

Id-kort

Når et klientsystem kalder en webservice, skal webservicen identificere (autentificere) og godkende (autorisere) brugeren, inden webservicen kan returnere de ønskede fortrolige oplysninger.

Denne autentifikation og autorisation sker på baggrund af oplysningerne i et SOSI id-kort i SOAP-headeren. Id-kortets oplysninger sammenholdes med webservicens egne oplysninger om godkendte brugere, f.eks. oplysninger fra et register, en spærreliste eller på baggrund af tillid til den instans, der har signeret id-kortet. Id-kortet benyttes til autentifikation af brugeren (enten en person eller et it-system).

Id-kortet udfyldes af klientsystemet og indeholder bl.a.:

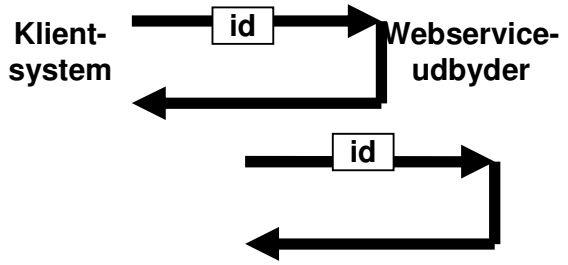
- brugerens CPR-nummer – eller andet id
- id-kortets udløbstidspunkt – normalt 24 timer efter kortets udstedelse
- eventuelt yderligere brugerdata i form af navn, e-mailadresse, rolle o.l.
- eventuel elektronisk signatur af id-kortets data.

<table>
<tr><td colspan="2">Id-kort for (Subject name ID): 2606444917</td></tr>
<tr><td>Kort udsteder: TDCHealth
Kort ID:AAATX
Kort type (System el. medarbejder):user
Kort version:1.0
Kort autentifikationsniveau (1-4): 4</td><td>Udstedt: 01-06-2006 Kl. 07:53:00
Gyldigt fra: 01-06-2006 Kl. 08:00:00
Gyldigt til: 01-07-2006 Kl. 07:53:00</td></tr>
<tr><td colspan="2">IT-systemoplysninger:</td></tr>
<tr><td>IT-systemets ID:LægeSystemA</td><td>Organisationens ID:079741
(ID format: medcom:ynumber)
Organisationens navn: Lægehuset, Vandværksvej.</td></tr>
<tr><td colspan="2">Evt. brugeroplysninger:
CPR-nummer: 2606444917
Stilling: Læge
Fornavn: Ole H.
Efternavn: Berggren
E-mail: ohb@nomail.dk
Evt. autorisationsnummer: 24778
Bruger rolle: PRAKTISERENDE_LAEGE</td></tr>
<tr><td colspan="2">Sikkerhedsniveau 2:
Username: ohb
Password: ohbPaWW5</td></tr>
<tr><td colspan="2">Sikkerhedsniveau 3: VOCES virksomhedssignatur
DigestValue: G3cubVicjk36Xj0IfyCjU0L11wE
SignatureValue: PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhk etc…
X509Certifikat: gAwIBAgIEQDZLNzANBg etc…
sosi:OCESCertHash": ALiLaerBquie1/t6ykRKqLZe13Y</td></tr>
<tr><td colspan="2">Sikkerhedsniveau 4: MOCES medarbejdersignatur
DigestValue: G3cubVicjk36Xj0IfyCjU0L11wE
SignatureValue: PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhk etc…
X509Certifikat: gAwIBAgIEQDZLNzANBg etc…
sosi:OCESCertHash": ALiLaerBquie1/t6ykRKqLZe13Y</td></tr>
</table>

Id-kortet anvendes således i situationer hvor webserviceudbyderen selv står for autentifikationen af brugeren:

1. Klientsystemet udsteder et id-kort til brugeren eller systemet.
   - Hvis id-kortet er af typen "user", *skal* brugerens CPR-nummer indsættes i to felter i id-kortet:
     - i "Subject@NameID"
     - i CPR-feltet under "brugeroplysninger".
   - Hvis id-kortet er af typen "system", *skal* it-systemets navn indsættes i to felter i id-kortet:
     - i "Subject@NameID"
     - i ITSystemName-feltet under "systemoplysninger".
   - Kortets udstedelsestidspunkt, ikrafttrædelsestidspunkt og udløbstidspunkt indsættes.
   - Yderligere brugerdata indsættes.

SOSI id-kort



2. Id-kortet signeres elektronisk med et MOCES eller VOCES certifikat af klientsystemet, hvis sikkerhedsniveauet kræver det.
   - Brugeren autentificeres (promptes for password). For systemer hentes password andetsteds fra.

3. Id-kortet medsendes i DGWS-request til webserviceudbyderen.

4. Webserviceudbyderen validerer id-kortet:
   a. Webserviceudbyderen kontrollerer de medsendte akkreditiver.
   b. Webserviceudbyderen kontrollerer id-kortets udløbstidspunkt.
   c. Kaldet afvises, hvis kortet er udløbet (ældre end 24 timer).
   d. Kaldet afvises, hvis kortet er ældre end den timeout, der kræves ved den aktuelle webservice (henholdsvis 5 min., 30 min. eller 8 timer).

5. Webserviceudbyderen autoriserer brugeren ved om muligt dels at checke at de medsendte oplysninger er korrekte og dels ved at anvende dem til at afgøre om brugeren har ret til at kalde den valgte service.

6. Webserviceudbyderen tager en kopi af id-kortet. Kopien kan bruges til at genkende brugeren ved opkald til samme webservice.

7. Id-kortet medsendes uændret i efterfølgende forespørgsler til samme webserviceudbyder, indtil id-kortet udløber efter 24 timer (eller den timeout, der kræves af den aktuelle webservice).

Single SignOn

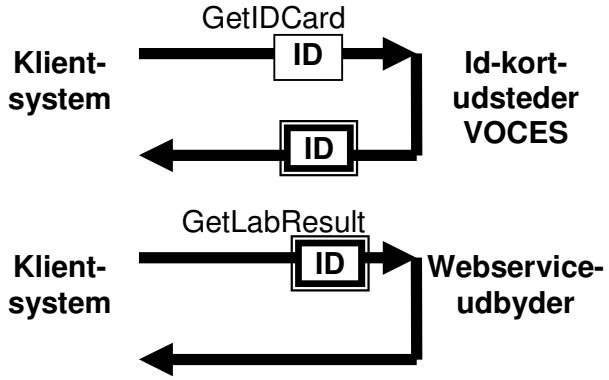
Den Gode Webservice er primært beregnet til direkte kommunikation mellem et klientsystem og en webserviceudbyder. I mange tilfælde vil der imidlertid være behov for, at et klientsystem skal kalde mange webserviceudbydere og autentificere sig over for hver enkelt. Her anvender Den Gode Webservice en særlig webserviceudbyder, der kaldes en identitetsservice (en "Identity Provider" eller "IdP"), til at foretage autentifikationen én gang for alle.

Når mange serviceudbydere danner et sådant netværk, der er baseret på tillid, kaldes det en "føderation". Når id-kortet er udstedt, kan det anvendes i 24 timer mod andre serviceudbydere i føderationen.

Et centralt udstedt id-kort vil kunne benyttes til at etablere en fælles "Single SignOn"-løsning i sundhedssektoren, på samme måde som dette i dag kendes ved anvendelse af den digitale TDC-signatur for privatpersoner mod webløsninger.

Afhentning af id-kort

1. Klientsystemet danner et id-kort og signerer det med sin digitale OCES medarbejder-signatur (MOCES).
2. Klientsystemet logger på den centrale identitets server, der verificerer MOCES-signaturen og tjekker spærrelisten.
3. IdP'en overskriver id-kortets oplysninger med verificeret CPR-nummer og nye tidsstempler.
4. IdP'en beriger evt. id-kortet med brugeroplysninger
5. IdP'en fjerner akkreditiverne fra id-kortet og erstatter disse med egen VOCES-signatur og eget certifikat. IdP anvender altid VOCES-signatur.
6. IdP'en returnerer det VOCES-signerede id-kort i responsemeddelelsen.
7. Klienten checker at VOCES signaturen er OK, dvs. at den er lavet med IdP'ens certifikat.
8. Klientsystemet vedlægger en kopi af det VOCES-signerede id-kort i fremtidige webservicekald, som webservicen kan sammenholde med sin egen kopi.



Ved SingleSignon vil identitetsservicen validere id-kortet, fjerne de gamle akkreditiver og selv underskrive id-kortet med sin egen private VOCES-nøgle, inden det returneres til klienten.

Et autentificeret id-kort vil altså aldrig indeholde brugernavn og password eller MOCES-signatur, men altid en VOCES-signatur fra den identitetsudbyder, der foretog autentifikationen.

Ved anvendelse af autentificerede id-kort er det et krav, at den der skal validere signaturen på forhånd er i besiddelse af certifikatet. Dette forhindrer at en tredjepart ændrer oplysningerne i id-kortet og erstatter signatur og certifikat med falske værdier. Man kan derfor med fordel nøjes med at indsætte en reference til det certifikat, som man kan validere underskriften med i KeyInfo/KeyName-feltet. Referencen angives som certifikatets cvr-rid værdi, der genbruges selvom certifikatet fornys.

OCES digital signatur

I Den Gode Webservice anvendes digital signatur to steder:

- **Signering af id-kortet**. Denne signering sikrer, at id-kortets oplysninger er korrekte.
- **Signering af hele SOAP-meddelelsen**. Denne signering sikrer, at hele meddelelsen ikke er ændret, og at den kommer fra den rigtige afsender. Dette kaldes også uafviselighed.

I begge tilfælde anvendes OCES digital signatur fra TDC.

Klientsystemet skal kunne håndtere forskellige metoder for at kunne anvende den digitale signatur.

Klientsystemet signerer sin meddelelse ved:

1. at kanonisere meddelelsen ved hjælp af C14N-kanoniseringsmetoden.
2. at udregne en digest (et "fingeraftryk") af den kanoniserede meddelelse ved hjælp af SHA1 digest-metoden.
3. at signere digesten ved hjælp af RSA-krypteringsmetoden.
4. at konvertere signaturen til ASCII-tegn ved brug af base64-metoden.

Når webservicen modtager meddelelsen, skal den:

1. base64-dekode signaturen, så den krypterede signatur kommer frem igen.
2. RSA-dekryptere signaturen, så den oprindelige hash-værdi kommer frem.

Metoder der benyttes ved brug af digital signatur

C14N-kanonisering
XML kan skrives på flere måder og stadig have samme betydning, f.eks. ved at lave mellemrum mellem tags eller ved at anvende en forkortet form for tags uden indhold, f.eks.
 i stedet for
</br> o.l. Disse forskelle fjernes ved "kanonisering", der i Den Gode Webservice anvender C14N-kanoniseringsmetoden.

SHA1-digest-udregning
Det egentlige fingeraftryk (kryptografisk digest eller hash-værdi) af den kanoniserede udgave af kilden sker ved at anvende SHA1-digest-metoden, der laver en digest med følgende væsentlige egenskaber:

1) De har altid en fast længde på 160 bytes uanset kildens størrelse.
2) Den samme besked giver altid den samme digest-værdi.
3) To forskellige beskeder giver altid forskellige digest-værdier.
4) Man kan ikke genskabe kilden fra digest-værdien.

RSA-signering (kryptering)
Det udregnede digest signeres (krypteres) med en hemmelig, privat nøgle. I OCES

anvendes nøgler af RSA-typen, hvorfor signeringsmetoden kaldes ”RSA-SHA1”.

Base64-konvertering
Både digest og signatur er krypteret og indeholder derfor en lang række specialtegn. For at sikre en uændret kommunikation af disse værdier konverteres begge til ASCII-tegnsættet ved brug af ”base64”-konvertering, inden de indsættes i XML-koden.

De metoder, der benyttes til kanonisering, udregning af hashværdien (”digesten”) og signeringen (krypteringen), fremgår af XML-elementerne <ds:CanonicalizationMethod>, <ds:DigestMethod> og <ds:SignatureMethod>. Metoderne er nærmere beskrevet i bilagene

Afsenderens signering og modtagers verificering af en OCES digital signatur foregår sådan:

Afsender-signering sker sådan:

- **Opret <ds:Signature> XML-elementerne** – som de fremgår af XML-listen i bilagene.
- **Udpeg det XML, der skal underskrives**. Dette gøres ved at angive enten ”#IDCard” eller ”#Envelope” i <ds:Reference URI="XXX"> elementet, alt efter om det er id-kortet eller hele SOAP-meddelelsen, der skal signeres.
- **Beregn et ”fingeraftryk”** (også kaldet et ”digest” eller en hash-værdi) af den XML-kode, der skal underskrives. Dette gøres sådan:
    - Selve signaturen signeres ikke. Derfor skal hele <ds:Signature>-elementet fjernes, før ”fingeraftrykket” beregnes.
    - Kanoniser den udpegede XML-kode efter C14N-metoden.
    - Hash-værdien af den kanoniserede XML-kode beregnes efter SHA1-metoden. Hash-værdien har altid en længde på 160 bytes.
    - Den udregnede hash-værdi base64-konverteres til ASCII-tegn.
    - Resultatet indsættes i <ds:DigestValue>-elementet.
- **Underskriv <ds:SignedInfo>-elementet.** Signeringen omfatter hele <ds:SignedInfo>-elementet, ikke kun den beregnede hash-værdi. Signeringen foregår sådan:
    - Kanoniser <ds:SignedInfo>-elementet ved hjælp af C14N-metoden.
    - Lav en digest (hash-værdi) af den kanoniserede XML-kode ved hjælp af SHA1-metoden.
    - Krypter de 160 bytes med den private nøgle, der hører til OCES-certifikatet ved hjælp af ”RSA-SHA1”-metoden.
    - Base64-konverter de krypterede bytes til ASCII-tegn.
    - Resultatet indsættes i <ds:SignatureValue>-elementet.

- **Gem OCES-certifikatet i <ds:KeyInfo>** (base64-kodet) i <ds:X509Certificate> elementet. Hermed får modtageren også afsenderens offentlige nøgle til dekryptering af den digitale signatur. Modtageren må dog aldrig blindt stole på certifikatet, men skal etablere tillid til det f.eks. ved at validere at det er et MOCES fra en kendt organisation.

- **Indsæt <ds:Signature> i den oprindelige XML-kode**.

Modtager-verificering sker sådan:

For at validere signaturen skal modtageren anvende stort set samme mekanismer, som når den digitale signatur bliver dannet:

1) **Valider <ds:Reference>**
    a. Udpeg det XML, hvis signatur skal valideres (id-kortet eller hele kuverten).
    b. Transformér id-kortet/kuverten ved at fjerne den indsatte signatur fra XML (enveloped transform).
    c. Transformér id-kortet/kuverten ved at "kanonisere det" med C14N-algoritmen.
    d. Beregn et SHA1-"fingeraftryk" (digest) af det transfomerede id-kort.
    e. Base64-dekod <ds:DigestValue> og sammenlign med det beregnede "fingeraftryk". De skal være ens.
2) **Valider <ds:SignatureValue>**
    a. Kanonisér <ds:SignedInfo>-elementet med C14N-algoritmen.
    b. Beregn et SHA1-"fingeraftryk" af det kanoniserede <ds:SignedInfo>-element.
    c. Base64-dekod OCES-certifikatet i <ds:KeyInfo> og find den offentlige nøgle.
    d. Dekrypter værdien af <ds:SignatureValue> med den offentlige nøgle.
    e. Sammenlign det beregnede "fingeraftryk" med det dekrypterede.
        - Hvis disse hashværdier er ens
            - er meddelelsen uændret
            - er afsenderen korrekt.

3) Man kan nu yderligere tjekke gyldigheden af certifikatet ved at aflæse dets udløbsdato, checke om det er på en spærreliste, validere at det er et OCES certifikat dvs. signeret af TDCs CA og at det er udstedt til en kendt organisation der har adgang (og ikke f.eks. det lokale autoværksted). Hvis signaturen også er i orden, kan man stole på den signerede information.

Det data, der benyttes ved den elektroniske signering, findes i SOAP-headeren i XML-elementet <ds:Signature>, hvor:

- <ds:Reference URI="#Envelope"> angiver, at det er hele SOAP-kuverten, der signeres. Hvis kun id-kortet signeres, angives i stedet "#IDCard".
- <DigestValue> er det udregnede "fingeraftryk" svarende til XML-koden.
- <ds:SignatureValue> er den medsendte digitale signatur af hashværdien (dvs. den krypterede hashværdi).
- <ds:X509Certificate> indeholder brugerens medsendte offentlige nøgle. Denne benyttes af modtageren til at dekryptere den signerede hashværdi.

<ds:Signature> XML for digital signatur

Den digitale signatur indeholder tre grupper data:

- <ds:SignedInfo> indeholder oplysninger om, hvad der skal signeres.
- <ds:SignatureValue> indeholder selve den digitale signatur.
- <ds:KeyInfo> indeholder den offentlige nøgle, modtageren skal bruge til at verificere signaturen.

```
<ds:Signature>

 <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="HTTP://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <ds:SignatureMethod Algorithm="HTTP://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#IDCard">
   <ds:Transforms>
    <ds:Transform Algorithm="HTTP://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    <ds:Transform Algorithm="HTTP://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="HTTP://www.w3.org/2000/09/xmldsig#sha1"/>
   <ds:DigestValue>G3cubVicjk36Xj0IfyCjU0L11wE=</ds:DigestValue>
  </ds:Reference>
 </ds:SignedInfo>

 <ds:SignatureValue>
  BaWKC9PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhkU9j9PvLQBIx+JdOYKGynzMRTJ8GqMJh6gh/cA2mgKJ9b
  qiNRVedxuw4/QnTYz0Yw/8kSO4X7MjdA7/pn0OwIDGCxkw3y4wJGLRR2dochINlFg=
 </ds:SignatureValue>

 <ds:KeyInfo>
  <ds:X509Data>
   <ds:X509Certificate>
    MIIFBDCCBG2gAwIBAgIEQDZLNzANBgkqhkiG9w0BAQUFADA/MQswCQYDVQQGEwJESzEMMAoGA1UENFUyBTeXN0ZW10Z
    XN0IENBIElJMB4XDTA1MDYwNjEyMDQwMDYwNjEyMzQwMFowfTELMAkGA1UEBhMCREsxLzAtBgNVBAoUJlREQyBUT1RB
    TEzYU05JMgLy8gQ1ZSOjI1NzY3NTM1MT0wFAYDVQQDEw1UZXN0IEJydWdlciAyMCUGA1UEBRMeZSOjI1NzY3NTM1LVJ
    JRDoxMTE4MDYxMDQzMzU2MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBuvze+4T1i0inhmvafWB2d81q3AG7ds06eG
    y+eLjQYumaY5EViSv4qyNwmnV6Y1sVi3LpD//wr7+DBanwBUEXnlzRY4No4U3DrDAjvl4NKjdv/Dkg1pMfUwmaIYkQo
    LTWHe8bCfVPXtovQ12CLO7uydoBzTQIDAQABo4ICzTCCAskwDgYDVR0PAQH/BAQDAgP4MCsGA1UdEAQkMCKAMTIwNDA
    wWoEPMjAwNzA2MDYxMjM0MDBaMEYGCCsGAQUFBwEBBDowODA2BggrBgEFcDovL3Rlc3Qub2NzcC5jZXJ0aWZpa2F0Lm
    RrL29jc3Avc3RhdHVzMIIBAwYDVR0gMIH4MIH1BgkpAQEBAQEBAQIwgecwLwYIKwYBBQUHAgEWI2h0dHA6Ly93d3cuY
    2VydGlmaWthkay9yZXBvc2l0b3J5MIGzBggrBgEFBQcCAjCBpjAKFgNUREMwAwIBARqBl1REQyBUZXN0IENlEgdWRzd
    GVkZXMgdW5kZXIgT0lEIDEuMS4xLjEuMS4xLjEuxLjIuIFREQyBUZXN0IENlcnRpZmljYXRlcyBmcm9tIHRoaXMgQ0E
    gYXJlIGlzc3VlZCB1bmRlxLjEuMS4xLjEuMS4xLjEuMi4wGgYJYIZIAYb4QgENBA0WC2VtcGxveWVlV2ViMCAGUdEQQ
    ZMBeBFXN1cHBvcnRAY2VydGlmaWthdC5kazCBlgYDVR0fBIGOMIGLMFagVKBSpFAwTjELUEBhMCREsxDDAKBgNVBAoT
    A1REQzEiMCAGA1UEAxMZVERDIE9DRVMgU3lzdGVtdGVzdCBDUEAxMEQ1JMMjAxoC+gLYYraHR0cDovL3Rlc3QuY3JsL
    m9jZXMuY2VydGlmaWthkay9vY2VzLmNybDAfBgNVHSMEGDAWgBQcmAlHGkw4uRDFBClb8fROgGrMfjAdBgNVHQ4EFgQ
    UpQWIRbZKFhWkcHOi1bgdX4YwCQYDVR0TBAIwADAZBgkqhkiG9n0HQQAEDDAKGwRWNy4xAwIDw0BAQUFAAOBgQBp+zm
    RburdSGirxmMWFFcT4NaP3W+XRPqY3iCiZuW2FcBrTtHyuFrjbQHg9RznxAgHIpzu/txQsSqv+m76Ki8zB2+r0fwlYr
    ABvcloPUfRF6pRksYtYNXsnGSRe1147c9K315hXG3QMmuU+rBFyvRGkWx0wIf3lOrLg==
   </ds:X509Certificate>
  </ds:X509Data>
 </ds:KeyInfo>
</ds:Signature>
```

VPN- og SSL-kryptering

Når man benytter Den Gode Webservice, er den digitale signering og kryptering adskilt i to processer:

- Først signeres dokumentet som en integreret del af klientsystemets øvrige funktionalitet.

- Når meddelelsen er færdig og skal sendes, krypteres kommunikationslinjen ved hjælp af et af internettets to standardmekanismer til at sikre transportlaget, SSL eller VPN:

    - SSL-kryptering (Secure Socket Layer) benyttes ved kommunikation over det åbne internet fra en bestemt applikation (f.eks. en webbrowser) til en bestemt udbyder (f.eks. Sundhed.dk).
    - VPN giver hele klientmaskinen sikker adgang til et andet netværk, f.eks. MedComs sikre sundhedsdatanet eller KMD-Net.

Fire timeouts

Webserviceudbyderen afgør på baggrund af en vurdering af den konkrete webservice, hvor ofte brugeren skal identificere sig – eller med andre ord hvor ofte klientsystemet skal genautentificere brugeren (prompte for password) og derefter udstede et nyt id-kort.

Ved indberetning af en dødsattest til Sundhedsstyrelsen er det f.eks. vigtigt, at det sikres, at kun den pågældende læge kan udfylde attesten. I disse tilfælde stilles der krav om, at lægen indtaster sit password ved hver indberetning (dvs. ved hvert HTTP-kald). I andre situationer kan det være tilstrækkeligt, at brugeren kun indtaster sit password hvert 30. minut eller hver 8. time.

I Den Gode Webservice anbefales, at webserviceudbyderne vælger mellem fire timeout-niveauer: ved hvert HTTP-kald, efter 30 minutter, efter 8 timer eller efter 24 timer.

Timeout-niveau		**Sæt kun et kryds**
4: Ved hvert HTTP-kald	Webserviceudbyder kræver fornyet bruger-autentifikation ved hvert nyt HTTP-kald. Id-kortet udløber efter 5 minutter.	
3: Efter 30 minutter	Webserviceudbyder kræver fornyet brugerautentifikation efter 30 minutters brug af id-kortet.	
2: Efter 8 timer	Webserviceudbyder kræver fornyet brugerautentifikation efter 8 timers brug af id-kortet.	
1: Efter 24 timer	Webserviceudbyder kræver fornyet bruger autentifikation efter 24 timers brug af id-kortet	

- Timeout-tidspunktet udregnes ud fra det udstedelsestidspunkt, der findes i SOSI-id-kortets udstedelsetidsstempel i XML-attributten <IssueInstant>. F.eks. vil et id-kort, der er udstedt for en time siden, ikke blive godkendt af webserveren ved timeout-niveau 4 eller 3.

- Såvel webserviceudbydere som klientsystemer skal sikre, at deres systemure går nøjagtigt ens, f.eks. ved at benytte samme tidservice via internettet.

- Klientsystemet skal sikre, at brugeren promptes for nyt password, tidsnok til at der kan udstedes og fremsendes et nyt id-kort til webserviceudbyderen, inden det gamle id-kort får timeout.

- Ved timeout tilbagesender webserviceudbyderen en fault-meddelelse til klientsystemet, der redegør for årsagen til timeout og forklarer, hvordan brugeren kan genoptage kommunikationen.

- Ved Single Signon indebærer timeout, at et nyt id-kort skal hentes hos den centrale IdP-kortudsteder, så en kommende timeout kan overholdes.

Fem sikkerhedsniveauer

Når to parter skal udveksle informationer, vil informationernes grad af følsomhed være afgørende for behovet for at bevise sin identitet. Er der tale om meget følsomme data, må der anvendes meget pålidelige mekanismer, mens mindre følsomme data ikke kræver så stor sikkerhed i autentifikationsprocessen.

Sikkerhedsbehovet er med andre ord defineret af, hvor meget tillid man kan have til en anden parts identitet, relateret til hvor meget tillid der er nødvendig.

F.eks. afkræves en bruger kun sit certifikatpassword hvert 30. minut på den elektroniske medicinprofil PEM, mens Sundhedsstyrelsen kræver, at en bruger autentificeres på ny ved hvert webservicekald, når lægen udfylder en officiel dødsattest over nettet.

Derfor indeholder Den Gode Webservice 5 forskellige sikkerhedsniveauer og 4 timeouts. Det højeste sikkerhedsniveau er niveau 5, hvor der skal anvendes digital signatur for hele SOAP-kuverten, og hvor brugeren afkræves password ved hvert nyt HTTP-kald.

5 sikkerhedsniveauer	
5 Digital_Signatur	Hele SOAP-meddelelsen OCES-signeret (uafviselig)
4 Medarbejder_ID signatur	SOSI-id-kort signeret med OCES-medarbejdercertifikat
3 System_ID signatur	SOSI-id-kort signeret med OCES-virksomhedscertifikat
2 Username_Password	Brugerkontrolleret i eget register
1 No_ID	Brugeridentifikation ikke nødvendig

- På alle sikkerhedsniveauer forudsættes det, at transportlaget er sikret ved hjælp af VPN- eller SSL-kryptering.
- På sikkerhedsniveau 5 anvendes en bindende underskrift (digital signering) på både request- og response-meddelelserne. Signeringen omfatter hele SOAP-konvolutten.
- For at kunne understøtte ”plug-and-play”-kommunikation på landsplan er det hensigten, at alle klientsystemer på sigt skal kunne understøtte alle fem sikkerhedsniveauer, mens det krævede sikkerhedsniveau på serveren afhænger af den aktuelle webservice.

Som nævnt fastlægger den enkelte webserviceudbyder, hvilken sikkerhedshåndtering klientsystemerne skal overholde, når de kommunikerer med webservicen. Webserviceudbyderen skal definere:

- **sikkerhedsniveau** (for identifikation af brugeren og signering af meddelelsen)
- **timeout-niveau** (hvor lang tid brugeren maksimalt må være på uden at skulle forny sit password).

Når webservice-systemet designes, skal webserviceudbyderen vælge en af de fem sikkerhedsniveauer.

- På sikkerhedsniveau 5 signeres hele SOAP-meddelelsen med OCES digitale signaturer for både request- og responsemeddelelser. For at det skal kunne lade sig gøre, skal både klientsystem og webserver kunne håndtere både afsendelse (signering) og modtagelse (verificering) af digital signatur.

- På sikkerhedsniveau 4 og 3 signeres kun det indeholdte SOSI id-kort.
  - På sikkerhedsniveau 4 autentificeres en medarbejder personligt.
  - På sikkerhedsniveau 3 autentificeres alene det pågældende it-system eller den pågældende organisation.

  Både niveau 4 og 3 kræver, at klientsystemet (eller en benyttet Identity Provider) kan håndtere afsendelsen (signeringen), og at webserveren kan håndtere modtagelsen (verificeringen) af den digitale signatur. Ved Single Signon henter klientsystemet id-kortet hos en Identity Provider.

- På sikkerhedsniveau 2 benyttes Username_Password til at sikre autentifikationen. Dette sikkerhedsniveau benyttes langt de fleste steder i dag.

- På sikkerhedsniveau 1 autentificeres brugeren ikke. Sikkerhedsniveau 1 kan i mange tilfælde være et tilstrækkeligt sikkerhedsniveau – også til kommunikation af personfølsomme data. Det anvendes f.eks. ved kommunikation af lab-opslag på det lukkede VPN-baserede sundhedsdatanet.

Ud over at vælge sikkerhedsniveau skal webserviceudbyderen også beslutte, hvilket timeout-niveau de ønsker – altså hvor lang tid der maksimalt må gå, inden brugeren skal autentificeres af webservicen igen.

Når niveau 5 anvendes sammen med et id-kort på niveau 3 eller 4, som er udstedt af en IdP, er det nødvendigt at sikre sig, at det certifikat der kan bruges til at validere signaturen på hele kuverten faktisk også er det, der blev brugt til autentifikation af IdP'en. Hvis dette ikke checkes er det nemlig nødvendigt for webserviceudbyderen selv at validere certifikatets gyldighed, da en uhæderlig 3.part ellers ville kunne erstatte certifikatet med et selvudstedt et, der f.eks. har de samme cvr-rid værdier indsat.

For at gøre det let for webserviceudbyderen at checke certifikatet uden at skulle validere det helt forfra, indeholder id-kortet XML-elementet med @name='sosi:OCSCertHash', der indeholder en base-64 kodet udgave af et sha-1 digest af det certifikat, der blev brugt til autentifikationen hos IdP'en. Ved at beregne en tilsvarende sha-1 hashværdi af det certifikat, der er medsendt på niveau og sammenligne de to værdier, kan webserviceudbyderen opnå tillid til signaturens gyldighed.

Id-kortets autentifikationsniveau

Id-kortets autentifikationsniveau skal fremgå af SOSI-Id-kortets ”Autentification Level”. I de fleste tilfælde vil autentifikationsniveauet være identisk med det sikkerhedsniveau, man har valgt. Det gælder for sikkerhedsniveau 1-4, mens man på sikkerhedsniveau 5 kan have et autentifikationsniveau på 1, 3 eller 4.

Autentifikationsniveau		**Sæt kun et kryds**
4:Medarbejder_ID	Brugers id verificeret med MOCES-medarbejdercertifikat	
3:System_ID	Systemets id verificeret med VOCES-virksomhedscertifikat	
2:Username_Password	Brugers id verificeret med Username_Password	
1:No_ID	Brugers id ikke verificeret	

Individuel anmodning om uafviselighed

Når en webservice stiller krav om sikkerhedsniveau 5 (”Digital Signatur”), sikres det, at både request- og response-kommunikationen er uafviselig. Det indebærer bl.a., at serviceudbyderen ved, hvilken sundhedsfaglig person der har haft adgang til eller opdateret en patients oplysninger.

Hvis man har et klientsystem på sikkerhedsniveau 2-4, er det dog også muligt at anmode om en kvittering for den efterfølgende response-meddelelse (”Digital Signatur”). Det gøres ved at indsætte ”yes” i XML-elementet <medcom:RequireNonRepudiationReceipt>

Om den efterfølgende response meddelelse rent faktisk signeres digitalt vil være afhængig af om den aktuelle webservice har implementeret signatur funktionaliteten.

Hvis modtageren ikke understøtter digital signatur, returneres en fault-fejlmeddelelse med følgende indhold:

```
<soap:Fault>
  <faultcode>Server</faultcode>
  <detail>
    <medcom:FaultCode>nonrepudiation_not_supported<medcom:FaultCode>
  </detail>
  <faultstring>Denne webservice understøtter ikke digital signatur. Fremsend en ny forespørgsel - men denne gang uden at anmode om et digitalt signeret svar</faultstring>
</soap:Fault>
```

Prioritet

Den Gode Webservice skal understøtte kommunikationen i sundhedssektoren. Her kan der opstå situationer, hvor det er afgørende, at en meddelelse kommer meget hurtigt frem til alle og dermed får prioritet. Prioriteringen angår såvel notifikationer og alarmer på brugerside som den tekniske prioritering og hastighed undervejs gennem it-systemer og -netværk.

Man kan angive meddelelsens prioritet ved i SOAP-header at indføje AKUT, HASTER eller RUTINE i XML-elementet Priority, f.eks. sådan:

```
<medcom:Priority>RUTINE</medcom:Priority>
```

- Modtagersystemer opfordres til at implementere brugernotifikationer eller en alarm, når det modtager en akut meddelelse.
- Den enkelte bruger må forvente, at det ikke er alle servere eller klientsystemer, der er i stand til at fremskynde kommunikationen eller notificere brugere individuelt.

Statuskoder og kvitteringer

I Den Gode Webservice anvendes to slags response-statuskoder (kvitteringer): HTTP-statuskoden og header-statuskoden <FlowStatus>. Derudover kan der returneres en <Fault>-fejlmeddelelse, hvor afsenderen i egen kode og egen tekst beskriver kommunikationsproblemet for modtageren.

HTTP-statuskode

HTTP statuskoden angiver kun, om meddelelsen har kunnet behandles (kode 200) eller ej (kode 500).

Koden fremgår af de første linjer i den HTTP-header, der sendes umiddelbart inden SOAP XML-filen.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="UTF-8"?>
<!--MedCom Den Gode Websrvice Response-->
<soapenv:Envelope….
```

I Den Gode Webservice benyttes alene de to HTTP-statuskoder ”200” og ”500”:

HTTP-statuskoder i Den Gode Webservice	
HTTP-header	**Kodebetydning**
200 OK	Meddelelsen er modtaget. SOAP:Body indeholder et svar.
500_Internal_Server_Error	Meddelelsen er ikke processeret. SOAP:Body indeholder en SOAP:Fault

Flowstatus

Header-statuskoden <FlowStatus> er kun med i positive kvitteringer hvor HTTP statuskoden er 200. I negative kvitteringer med HTTP statuskode 500 indeholder soap:Fault-elementet fejlkoden i <medcom:FaultCode> under <detail> elementet.

- Hvis kommunikationen er forløbet tilfredsstillende, kvitteres positivt ved koden ”flow_running” eller koden ”flow_finalized_succesfully”:

Positive kvitteringer i Den Gode Webservice	
<FlowStatus>	**Kodebetydning**
flow_running	Webserviceudbyderen har modtaget og behandler den modtagne request. Klientsystemet skal kalde webserveren senere for at hente en efterfølgende response.
flow_finalized_succesfully	Webservicesessionen er afsluttet succesfuldt. Klientsystemet behøver ikke nødvendigvis at kalde webserveren igen. Hvis klientsystemet benytter sidste kald i en session flere gange, vil der blive returneret flere ”færdigbehandlet” kvitteringer.

- Hvis der er problemer med kommunikationen, kvitteres negativt med en af nedenstående koder eller en webservice specifik kode:

Negative kvitteringer i Den Gode Webservice	
<detail>	**Kodebetydning**
syntax_error	Beskeden indeholdt data, der ikke blev forstået af serveren.
missing_required_header	Der mangler en eller flere obligatoriske DGWS-headere i den medsendte besked, f.eks. id-kort, som altid skal være der.
security_level_failed	Autentifikation eller autorisationsfejl. Forkert valgt sikkerhedsniveau.
invalid_username_password	Autentifikation eller autorisationsfejl. Fejl i username/password
invalid_signature	Autentifikation eller autorisationsfejl. Fejl i digital OCES-signatur enten på id-kortet eller på hele kuverten.
invalid_idcard	Autentifikation eller autorisationsfejl. Fejl i SOSI id-kort, f.eks. at CPR-nummeret ikke matcher det, der kan slås op via certifikatet.
invalid_certificate	Autentifikation eller autorisationsfejl. Certifikat er ikke OCES, spærret eller udløbet.
expired_idcard	Autentifikation eller autorisationsfejl. SOSI-id udløbet eller for gammelt for denne webserviceudbyder.
not_authorized	Brugeren har ikke rettigheder til at udføre denne webservice.
illegal_HTTP_method	Bruges, hvis en klient sender alle andre HTTP Methods end ”GET” og ”POST”. Bruges også, hvis en server ikke udstiller webservice-implementation via ”GET”.
nonrepudiation_not_supported	Webserviceudbydersystemet kan ikke håndtere at lave en digital signatur på svaret.

Fault

Når der sendes en negativ kvittering i en response-besked, indsættes en fault-fejlmeddelelse, der konkret beskriver fejlsituationen, og hvad klientsystemet om muligt skal gøre for at undgå en lignende situation.

Fault-meddelelsen skal i givet fald være det eneste indhold i body:

```
<soap:Body>
  <soap:Fault>
    <faultcode>Server</faultcode>
    <detail>
      <medcom:FaultCode>invalid_idcard</medcom:FaultCode>
    </detail>
    <faultstring>Det angivne CPR-nummer matcher ikke certifikatets</faultstring>
  </soap:Fault>
<soap:Body>
```

- HTTP statuskoden har værdien 500.
- <faultcode> har standard SOAP fault værdien ”Server”, der angiver at der skete en fejl i behandlingen af kaldet.
- <detail> indeholder en af fejlkoderne fra listen ovenfor eller en webservice specifik fejlkode. Fejlkoder indlejres altid i medcom:FaultCode elementet inden i detail.

- <faultstring> indeholder en menneskeligt læsbar tekst, der giver hints om hvordan fejlen evt. kan udbedres. Teksten udarbejdes af den webserviceudbyder, der tilbyder den aktuelle webservice og angiver f.eks. hvilke felter der mangler at blive udfyldt, eller hvilke tekniske eller syntaksmæssige fejl den forudgående request-meddelelse indeholdt.
- Serviceudbyderen skal udarbejde en liste over de fault-koder og tilsvarende tekster, der benyttes i webservicen, som kan udleveres til klientsystemer.
- Klientsystemet skal sikre, at fejlene rettes, inden webservicen kaldes op igen. Man må ikke blive ved med automatisk at sende uændrede meddelelser.
- Fejlmeddelelsen kan vises for slutbrugeren eller en overvågningsfunktion.

Design af snitflader

De services og den tilhørende datamodel, en serviceudbyder udstiller, kaldes en snitflade.

Design af gode snitflader er ikke altid nemt. Et godt snitfladedesign giver en løs kobling fra det bagvedliggende system, dvs. at snitfladen er uafhængig af serviceudbyderens infrastruktur. Det medfører, at snitfladen er resistent over for forandringer i den bagvedliggende infrastruktur. På datasiden kan løs kobling realiseres ved at genbruge globalt definerede datatyper.

WSDL'en er kontrakten, der beskriver snitfladen mellem webserviceudbyder og en klient. Der findes to måder at frembringe en WSDL:

Kode først - fast kobling

Kontrakten genereres ud fra tidligere implementeret forretningslogik eller en model herfor, f.eks. udarbejdet i UML. Denne metode er besnærende, fordi den gør det meget let at lave en webservice, og måske derfor er det også den metode, der er bedst værktøjsunderstøttet i dag. Ulempen er, at ændringer i den bagvedliggende kode vil påvirke alle klienter, da snitfladen dermed ændres. Samtidig er det overordentlig vanskeligt at overholde profiler som f.eks. OIOXML.

Kontrakt først - løs kobling

Webservicekontrakten udarbejdes efter det behov, servicen skal dække. Hvordan selve servicen realiseres, er uvedkommende for dette arbejde, og der er derfor en opgave i at implementere servicen vha. bagvedliggende forretningslogik potentielt med helt andre snitflader. På denne måde gøres webservicekontrakten helt uafhængig af den bagvedliggende implementation og bliver derfor meget mere robust. Denne metode er mere teknisk krævende, da den kræver indsigt i WSDL-, XML- og XML-skemaer.

Den Gode Webservice er på linje med andre åbne standarder, der er designet til at understøtte en "løs kobling" mellem it-systemer.

Den Gode Webservice stiller ikke krav til, hvordan en part udfører sin opgave. Brug af backend-systemer og intern kommunikation er et lokalt anliggende.

SOAP-header

Den samlede SOAP-header i Den Gode Webservice ser således ud:

Den Gode Webservice SOAP-kuvert

Afsendt:01-06-2006 Kl. 08:01:00
Message ID:AMRRMD
Flow ID:AGQ5ZW
Flow Status: flow_running

Prioritet: RUTINE
Signering ønskes: no
Sikkerhedsniveau (1-5): 5
TimeOut (min): 5

ID Kort for (Subject name ID): 2606444917

Kort udsteder: TDCHealth
Kort ID:AAATX
Kort type (System el.medarbejder):user
Kort version:1.0
Kort autentifikationsniveau (1-4): 4

Udstedt: 01-06-2006 Kl. 07:53:00
Gyldigt fra: 01-06-2006 Kl. 08:00:00
Gyldigt til: 01-07-2006 Kl. 07:53:00

IT-system oplysninger:

IT systemets ID:LægeSystemA

*Organisationens ID:*079741
(ID format: medcom:ynumber)
Organisationens navn: Lægehuset, Vandværksvej.

Evt. bruger oplysninger:

CPR nummer: 2606444917
Stilling: Maskinarbejder
Fornavn: Ole H.
Efternavn: Berggren
eMail: ohb@nomail.dk

Evt. autorisationsnummer: 24778
Bruger rolle: PRAKTISERENDE_LAEGE

Sikkerhedsniveau 2:
Username: ohb
Password: ohbPaWW5

Sikkerhedsniveau 3: VOCES virksomhedssignatur
DigestValue: G3cubVicjk36Xj0IfyCjU0L11wE
SignatureValue: PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhk etc…
X509Certifikat: gAwIBAgIEQDZLNzANBg etc…
sosi:OCESCertHash": ALiLaerBquie1/t6ykRKqLZe13Y

Sikkerhedsniveau 4: MOCES medarbejdersignatur
DigestValue: G3cubVicjk36Xj0IfyCjU0L11wE
SignatureValue: PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhk etc…
X509Certifikat: gAwIBAgIEQDZLNzANBg etc…
sosi:OCESCertHash": ALiLaerBquie1/t6ykRKqLZe13Y

Sikkerhedsniveau 5: OCES signatur for hele kuverten
DigestValue: G3cubVicjk36Xj0IfyCjU0L11wE
SignatureValue: PQRD1vDyf6ttx4/OKqP7I4TEm8m0B2AVV4O4OTGHWhk etc…
X509Certifikat: gAwIBAgIEQDZLNzANBg etc…

Body – brevet

Headeren indeholder tekniske forsendelsesdata, et id-kort og den digitale signatur for hele SOAP-kuverten.

- Alle de viste forsendelsesdata, id-kortets tekniske data og data om det afsendende it-system fremgår altid af alle meddelelser.
- Hvis id-kortet er af typen ”user”, fremsendes CPR-nummer og navn altid. Det er valgfrit, om der skal fremsendes flere brugeroplysninger.
- Data for de forskellige sikkerhedsniveauer fremsendes kun, hvis det enkelte sikkerhedsniveau anvendes. I så fald skal alle de viste data være udfyldt.
- CPR-nummer fremsendes i OiO-format (det vil sige uden bindestreg – men vises med bindestreg).
- Tidsangivelser fremsendes i amerikansk format, men vises som her i dansk format

Kuvertdata

Kuvertdata er tekniske og logistiske data, der vedrører forsendelse af hele meddelelsen. Den Gode Webservice indeholder følgende kuvertdata i SOAP-headeren:

- Afsendt (TimeStamp): Reelt afsendelsestidspunkt for beskeden.
- Message-id (MessageID): Unikt id for den pågældende meddelelse. Genbruges dog, hvis den samme besked sendes igen.
- Flow-id (FlowID): Unikt id for den pågældende session. Er identisk for alle beskeder i samme session og genbruges uændret ved genfremsendelse.
- Flow Status (FlowStatus): Positive eller negative kvitteringer (statuskoder). Se særskilt liste i bilagene.
- Prioritet (Priority): Besked til webserviceudbyder og klientsystem om, hvordan beskeden skal prioriteres: AKUT, HASTER eller RUTINE.
- Signering ønskes (RequireNonRepudiationReceipt): Klientens ønske om efterfølgende signeret response: ”no” eller ”yes”.
- Sikkerhedsniveau (SecurityLevel): Anvendt sikkerhedsniveau er 1- 5. 5 = ”Digital_Signatur”, 4 = ”Medarbejder_ID”, 3 = ”System_ID”, 2 = ”Username_Password” og 1 = ”No_ID”.
- Timeout (TimeOut): Angivelse af, hvor lang tid id-kortet er gyldigt efter udstedelse uafhængigt af valgt sikkerhedsniveau: 4: ”instant”, 3: ”30_Minuttes”, 2: ”8_Hour” og 1: ”24_Hour”.

Id-kort-data

Næsten alle sikkerhedsdata er samlet i et XML-element, der kaldes et SOSI-id-kort. Id-kortet indeholder følgende information:

- Bruger-id (Subject nameID): Identifikation af den person id-kortet er udstedt til.
- Kortudsteder (Issuer): Identifikation af den der har udstedt id-kortet.
- Kort-id (IDCardID):Unikt id for dette id-kort.
- Korttype (IDCardType): Angiver om id-kortets subject er en medarbejder eller et system: USER eller SYSTEM.
- Kortversion (IDCardVersion): Den version af id-kort-formatet, dette kort anvender.

- Kort-autentifikationsniveau (AuthenticationLevel): Lovlige værdier er 1-4. 1 = ingen autentifikation, 2 = brugernavn/password, 3 = VOCES-signatur, 4 = MOCES-signatur.
- Udstedt (IssueInstant): Dato og klokkeslæt for id-kortets udstedelse. Bruges som udgangspunkt for beregning af TimeOut.
- Gyldigt fra (Conditions NotBefore): Tidsstempel, der afgrænser SOSI id-kortets ikrafttrædelsestidspunkt.
- Gyldigt til (Conditions NotOnOrAfter): Tidsstempel, der afgrænser SOSI-id-kortets udløbstidspunkt.

Obligatoriske system- og organisationsoplysninger.
- It-systemets id (ITSystemName): Id (= navn) på det it-system, medarbejderen benytter til Den Gode Webservice-kommunikation. Skal også angives i "Name ID", når korttypen er et "system".
- Organisationens id (CareProviderID): Id for medarbejderens organisation.
- Organisationens navn (CareProviderName): Navn på medarbejderens organisation.

Eventuelle brugeroplysninger
- CPR-nummer (UserCivilRegistrationNumber): CPR-nummer eller anden id for id-kortets indehaver. Skal også angives i "Name ID", når korttypen er "user".
- Stilling (UserOccupation): Medarbejderens stillingsbetegnelse.
- Fornavn (UserGivenName): Medarbejderens fornavn(e).
- Efternavn (UserSurName): Medarbejderens efternavn.
- eMail (UserEmailAddress): Medarbejderens e-mailadresse.
- Brugerrolle (UserRole): Den funktion som medarbejderen er logget på under. Benyttes til brugerautorisation.
- Autorisationsnummer (UserAuthorizationCode): Medarbejderens sundhedsfaglige autorisationsnummer fra Sundhedsstyrelsen.

Ved sikkerhedsniveau 2 medsendes i XML-elementet "OCESSignature"
- Username (Username): Brugerens indtastede brugernavn.
- Password (Password): Brugerens indtastede password.

Ved sikkerhedsniveau 3 og 4 medsendes i XML-elementet "OCESSignature"
- DigestValue: Fingeraftrykket (hash-værdien) for id-kortet.
- SignatureValue: Den elektroniske signatur for id-kortet.
- X509Certificate: Brugerens TDC OCES-certifikat. Indeholder den offentlige nøgle.

Ved sikkerhedsniveau 5 medsendes i XML-elementet "OCESSignature2"
- DigestValue fingeraftrykket (hash-værdien) for hele kuverten.
- SignatureValue: Den elektroniske signatur for hele kuverten.
- X509Certificate: Brugerens TDC OCES-certifikat. Indeholder den offentlige nøgle, som afsenderen skal bruge ved dekryptering.

BILAG

Bilag 1: XML digital signering

Bilag 2: Id-kortet: Sådan bruges SAML-standarden

Bilag 3: Usecase-eksempler

Bilag 4: Dataliste

Bilag 5: Enumerations-liste

Bilag 6: WSDL for Den Gode Webservice

Bilag 7: XML-Liste

Bilag 8: Testeksempler

Bilag 9: XML Schema for Den Gode Webservice