

# **Teknisk Proof of Concept for Den fællesoffentlige integrationsmodel for borgerportalen og virksomhedsportalen**

## **Offentlig Erfaringsrapport**

# Indhold

Forord .....	3
1 Summary.....	4
2 Indledning .....	6
2.1 Baggrund .....	6
2.2 Projekt mål .....	7
2.3 Afgrænsninger .....	7
2.4 Projektfremgangsmåde .....	8
3 Opsætning, deltagere og cases i PoC .....	10
3.1 Borger.dk portalsiden: Oracle portal .....	10
3.2 Sundhed.dk: BMI selv-service .....	13
3.3 Færdselsstyrelsen: Vognmandstilladelse .....	15
3.4 KMD: Mit budget.....	18
3.5 IBM: CVR service .....	20
3.6 Rødovre Kommune: " e-posthus.dk" .....	23
4 Afprøvning og erfaringer .....	27
4.1 Generelle erfaringer og anbefalinger .....	27
4.2 Link eller iFrame m/Single Sing-on (SSO) .....	29
4.3 Remote Portlet – WSRP.....	34
4.4 Composite Application - JSR168 portlets .....	40
5 Konklusion .....	44

## **Forord**

Det tekniske Proof of Concept i regi af den fællesoffentlige integrationsmodel for borgerportalen og virksomhedsportalen har kun kunnet gennemføres med deltagelse af en række myndigheder og leverandører. Derfor stor tak for indsatsen og værdifulde input til:

- Færdselsstyrelsen
- IT- og Telestyrelsen
- Rødovre Kommune
- Sundhed.dk
- IBM
- KMD
- NNIT

I øvrigt tak til de mange myndigheder og interessenter, der har taget sig tid til at gennemlæse udkast til rapport og har afgivet hørings svar. Alle input er efter bedste evne vurderet og i mange tilfælde indarbejdet i rapporten.

God læsning!

Med venlig hilsen

Kristian Hjort-Madsen  
Forretnings- og it-arkitekt, Den Digitale Taskforce

# 1 Summary

Denne rapport samler op på resultater og erfaringer fra det tekniske "Proof of Concept" (herefter PoC) under den fællesoffentlige integrationsmodel for borgerportalen og virksomhedsportalen.

Grundlæggende er afprøvet de fire integrationsformer (links og frame her behandlet under ét) som Gartner har peget på i rapporten "Arkitekturen på en borgerportal" udarbejdet for MVTU.

Alle integrationsformer op mod portalen har virket under afprøvning. Så grundlæggende kan alle integrationsformer anvendes fra 2008 med de nuværende standarder og produkt-understøttelser. Der er dog fordele og ulemper ved alle integrationsformerne. Nærmere omkring resultater og erfaringer er beskrevet i afsnit 4.

Links er klart den mest simple integrationsform, men tilbyder ikke megen "integration" og brugen af SAML er ganske nyt for de fleste leverandører og ikke mindst danske implementeringspartnere. Der er således meget få erfaringer med at arbejde med SAML og Single Sign-on løsninger baseret på fælles logintjenester. Der er desuden en tidsmæssig udfordring, idet der for nuværende endnu ikke er etableret en fælles logintjeneste. Erfaringer i PoC er alene baseret på en logintjeneste opsat til afprøvningsformål.

Portlets (JSR168) er en velafprøvet standard, men så snart man bevæger sig på tværs af platforme og myndighedsinstallationer kan billedet kompliceres. Microsoft og .NET er i den sammenhæng helt ude, da det er en javastandard. JSR portlets er samtidigt den integrationsform, der er den *mindst* ønskværdige i forhold til "glaspladeprincippet" og en klar opdeling mellem myndigheds- og portalejers ansvar.

Remote Portlets (WSRP) er en relativ uafprøvet standard og der forventes indenfor overskuelig tid en ny version. Der er ved brug af WSRP behov for taktiske sikkerhedsløsninger (som dog ligner hvad der findes i dag i kørende portalløsninger).

Nedenfor opsummeres nogle retningslinier for i hvilke sammenhænge integrationsformerne forekommer mere eller mindre oplagte.

## Genbrug af eksisterende løsninger

Links og iFrame er nærliggende til at videreføre eksisterende web løsninger. Bemærk dog at brugen af SSO teknologi ikke er helt trivial.

Har man en java platform vil JSR168 og WSRP portlets oftest også være en oplagt mulighed. Har man derimod udviklet på Microsoft kan JSR udelukkes mens såvel link, iFrame som WSRP fortsat er kandidater. WSRP vil dog her være lidt mindre trivial, da Microsoft ikke umiddelbart understøtter WSRP som producer, hvorfor der skal anvendes 3. parts komponenter.

### **Tæt visuel integration?**

Hvis der ønskes en tæt visuel integration mellem myndighed og portal og evt. med services fra forskellige myndigheder bør der anvendes iFrames eller Remote Portlets. JSR168 portlets har samme mulighed for tæt visuel integration, jf. dog de allerede nævnte forbehold omkring glaspladeprincippet.

Er der modsat ønske om at afsender fortsat har egen identitet og brugeren får et tydeligt indtryk af at bevæge sig mellem forskellige hjemmesider er Link m/SSO oplagt.

### **Interaktion mellem myndighedsservices**

Er det behov for at forskellige services på portalen taler sammen, er det afgørende om man kan vente på WSRP 2.0 som forventes at understøtte dette.

Såfremt man har behov for interaktion *mellem* services fra forskellige myndigheder her og nu er der kun JSR168 portlets (eller i PoC-sammenhæng uafprøvede work arounds udenfor standarderne).

Myndigheder kan dog – uafhængigt af portal integration – altid opbygge serviceorienterede løsninger, hvor myndighederne selv implementerer en integration og *herefter* integrerer slutresultatet op mod en af portalerne. Det er hvad case'en med Færdselsstyrelsen viser.

### **Sikkerhed**

Såfremt sikkerhed er et væsentligt emne for en integration, skal man afklare om en taktisk løsning – uden brug af SAML tokens som viderefører det fælles login - er tilstrækkelig. Ellers skal man basere løsningen på Link eller iFrame med SSO indtil videre. Det forventes at ændre sig inden for kort tid hvor man må vente at en SAML profilering til andre services er klar.

Bemærk at afsnit 4 indeholder en opsummering af fordele og ulemper for hver integrationsform (i slutningen af hvert underafsnit).

Der er desuden gennem rapporten fremhævet en række konkrete opmærksomhedspunkter og anbefalinger, som er visuelt fremhævet i en grå boks. Disse kan læses uden at nærlæse den omkringliggende tekst.



## 2 Indledning

Denne rapport samler op på resultater og erfaringer fra det tekniske "Proof of Concept" (herefter PoC) under den fællesoffentlige integrationsmodel for borgerportalen og virksomhedsportalen.

Grundlæggende er afprøvet de fire integrationsformer (links og iFrame her behandlet under ét) som Gartner har peget på i rapporten "Arkitekturen på en borgerportal" udarbejdet for MVTU. Se evt. mere herom under fremgangsmåde.

Afsnit 2 beskriver i det følgende baggrund for PoC projektet, projektmål, afgrænsninger og fremgangsmåde. Afsnit 3 beskriver de cases og integrationsformer, der er anvendt i afprøvningen. Resultater beskrives herefter i afsnit 4 opdelt på generelle erfaringer samt specifikke for de enkelte integrationsformer. Endeligt rundes af med konklusioner i afsnit 5.

Der er gennem rapporten anvendt bokse, der fremhæver observationer eller forslag fra PoC projektet. Disse er markeret på følgende vis:

	Konkrete forslag eller "gode råd" fra PoC projektet – herunder ideer fra deltagende myndigheder og leverandører - til forfatterne af Integrationsmodellen.
	Væsentlige observationer eller særlige vinkler på en problemstilling. Ofte vil disse henlede opmærksomhed på en problemstilling, der ellers risikerer at blive overset eller undervurderet hos myndigheder eller leverandører.

### 2.1 Baggrund

Det tekniske "Proof of Concept" (herefter PoC) er en del af projektet for den fællesoffentlige integrationsmodel for borgerportalen og virksomhedsportalen.

Borgerportalen og virksomhedsportalen forventes at blive en såkaldt "smal" portaler, hvor ansvar for udvikling og drift af løsninger uforandret ligger hos myndighederne. De to portaler skal give borgerne adgang til overbliksinformation og selvbetjeningsløsninger samt give adgang til egne data fra alle offentlige myndigheder via en "Min Side". Myndighedernes selvbetjeningsløsninger og dataadgange skal integreres i Min Side via en åben og standardiseret Integrationsmodel.

Denne integrationsmodel vil ikke pege på én bestemt måde at integrere til portaler på, men omfatte en række "tilladte integrationsformer". Der er ud fra Gartner-rapporten "Arkitekturen på en borgerportal" udnævnt en række mulige integrationsformer som kandidater. Disse er i et vist omfang baseret på relativ ny og uafprøvet teknologi og er derfor underkastet afprøvning i PoC projektet.

## 2.2 Projekt mål

Projektet har haft til formål at afprøve en række integrationsformer, som er udnævnt som integrationskandidater og forventes at indgå i en samlet Integrationsmodel for, hvordan offentlige myndigheder kan eksponere selvbetjeningsløsninger og andet digitalt indhold på Borgerportalen og Virksomhedsportalen.

Projektet skal begrænse risikoen i et efterfølgende udviklingsforløb på disse portaler, som ellers risikerer at læne sig op af ”forkert teknologi” eller anvender den rette teknologi på en forkert måde. PoC'en har derfor fokuseret på de *mindst* velafprøvede teknologier og sikkerhedsmodeller samt de forventede høj-risiko områder.

En samlet Integrationsmodel for Borgerportalen og Virksomhedsportalen har til formål at indsnævre udfaldsrummet for, hvordan myndigheder eksponerer services/indhold på de to portaler på en fleksibel, standardiseret og økonomisk ansvarlig måde. Integrationsmodellen vil sikre en ensartet udvikling af offentlige services/indhold til de to portaler, og dermed reducere udviklings- og integrationsomkostningerne for såvel offentlige myndigheder som de to portaler. Modellen skal endvidere sikre øget genbrug af metoder og løsninger på tværs af den offentlige sektor igennem åbne standarder for integration. Nærværende rapport er et input blandt flere til den kommende integrationsmodel.

## 2.3 Afgrænsninger

PoC projektet er gennemført med en række bevidste fravalg – afgrænsninger. Der er foretaget følgende afgrænsninger i projektet, hvor af de to første var fastlagt allerede ved projektinitiering:

- **Alternative teknologier.** Der kan tænkes flere alternativer end de, der opstilles i Gartners rapport som danner grundlaget for de valgte integrationsformer. PoC har til formål at gennemføre en *praktisk afprøvning* af disse og ikke at ”udfordre” listen med yderligere alternative muligheder.
- **Web-clipping.** Gartner nævner Web-clipping som mulig integrationsform, men det afprøves ikke i PoC. PoC sætter primært fokus på de mindst afprøvede teknologier og standarder. Web-clipping findes og er i anvendelse og det er ikke prioriteret mht. afprøvning. Der har desuden ikke umiddelbart været politisk ønske om at understøtte Web-clipping på Borger- eller Virksomhedsportalen.
- **Performance.** Performance forventes ikke at indgå i PoC i form af egentlige belastningstests. Der gennemføres en vurdering af den konkrete integrationsform, i det omfang det er praktisk muligt.
- **Sikkerheds.** Der afprøves SSO sikkerhedsaspekter alene i link og iFrame sammenhæng, da der ikke er en profil, der angiver hvilke standarder vi bør afprøve i sammenhæng med WSRP og JSR for at opnå SSO.
- **Interaktion mellem myndighedsservices.** Der er ikke afprøvet portaldrevet kommunikation mellem myndighedsservices – eksempelvis inter-portlet kommunikation i JSR eller forsøg på tilsvarende. Der er ikke gjort forsøg på at vurdere WSRP 2.0 som integrationsform der kræver interaktion, men det nævntes i rapporten at interaktion forventes mulig.

Afprøvningen af Single Sign-on (SSO) med brug af fælles login-tjeneste blev begrænset til de områder, hvor der tegner sig et klart billede af, hvordan SSO med en fællesoffentlig logintjeneste vil skulle fungere. Indenfor Web Services og dermed Remote Portlets (WSRP) er der ikke et sådant klart billede – og dermed heller ikke umiddelbart "noget som kan afprøves" i PoC sammenhæng. Projektet har ikke forsøgt at omfatte egen profilering eller afprøvning af taktiske løsninger, men en *diskussion* af sidstnævnte har fundet sted.

## 2.4 Projektfremgangsmåde

### Deltagere

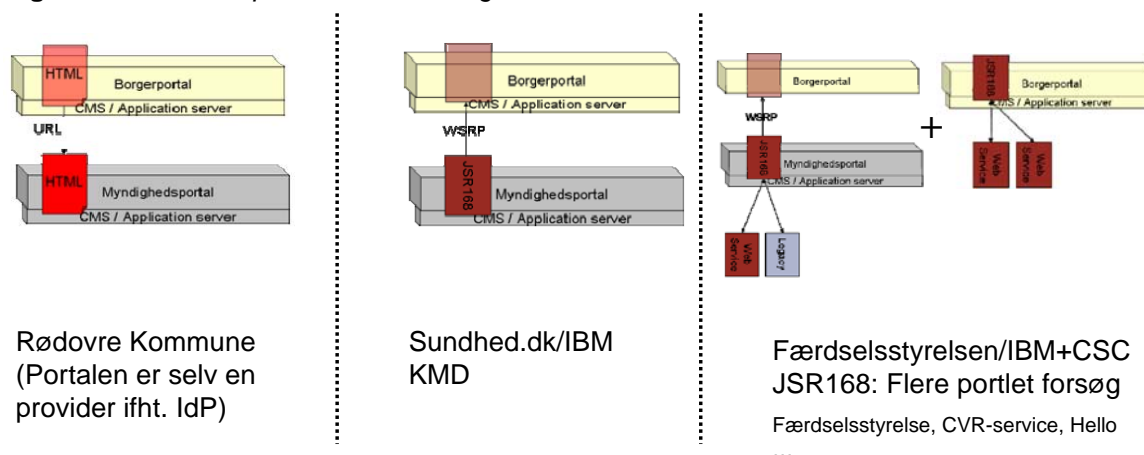
Der er gennemført et PoC, hvor såvel portal-siden (omtalt som consumer) som serviceprovider-/myndighedssiden (omtalt (service)provider) indgår sammen med en fælles brugerstyringsmodel.

Projektet har afprøvet forskellige integrationsformer i samarbejde med udvalgte myndigheder - de såkaldte "særtog" - og deres leverandører. For flere af disse samarbejdes der – af praktiske årsager - i høj grad direkte med den tekniske leverandør bag løsningen.

### Integrationsformer

Rapporten "Arkitekturen på en borgerportal" og dens anbefalinger lægges til grund for et PoC forløb, der således tager udgangspunkt i de integrationsformer der anbefales heri. Hver service fra en myndighed er implementeret i overensstemmelse med én integrationsform. Samme myndigheds case er således *ikke* afprøvet under flere forskellige integrationsformer. Følgende deltagere og integrationsformer er afprøvet:

**Figur 1: Én serviceprovider = én integrationsform**



### Brugeridentitet og fælles brugerstyring

Der har ikke været en oplagt kandidat til at levere fællesoffentlig brugerstyring indenfor de tidsmæssige rammer af PoC'en. Der er derfor etableret et setup, hvor de forventede standarder kan afprøves i sammenhæng med de udvalgte integrationsformer.

ITST har opstillet en Identity Provider til PoC projektet. ITST repræsenterer det offentlige holdning til, hvilke standarder der bør anvendes og Skat har værdifulde erfaringer fra det igangværende arbejde med implementering af model for brugerstyring - og har i dette projekt været med til at sikre, at vi ikke "går en anden vej" end det man siden ville kunne bygge på Skats kommende løsning (Skat kandiderer til at få drift af en kommende fællesoffentlig logintjeneste).

### **PoC miljø til endelig afprøvning**

PoC er gennemført på et adskilt miljø til netop dette formål, for ikke at komme i konflikt med andre igangværende udviklings- og/eller testforløb samt for at kunne afprøve en brugerstyringsmodel, der afviger fra hvad der ellers er i spil hos de involverede parter.

Der er dog grundlæggende anvendt samme teknologi som Borger.dk anvender i dag på portalsiden og den teknologi de enkelte myndigheder og leverandører bygger på i dag for at kunne drage anvendelige erfaringer ud af afprøvningen. Opdateringer af software er sket i det omfang, det har været nødvendigt og er medtaget som erfaringer i denne rapport.

## **3 Opsætning, deltagere og cases i PoC**

### **3.1 Borger.dk portalsiden: Oracle portal**

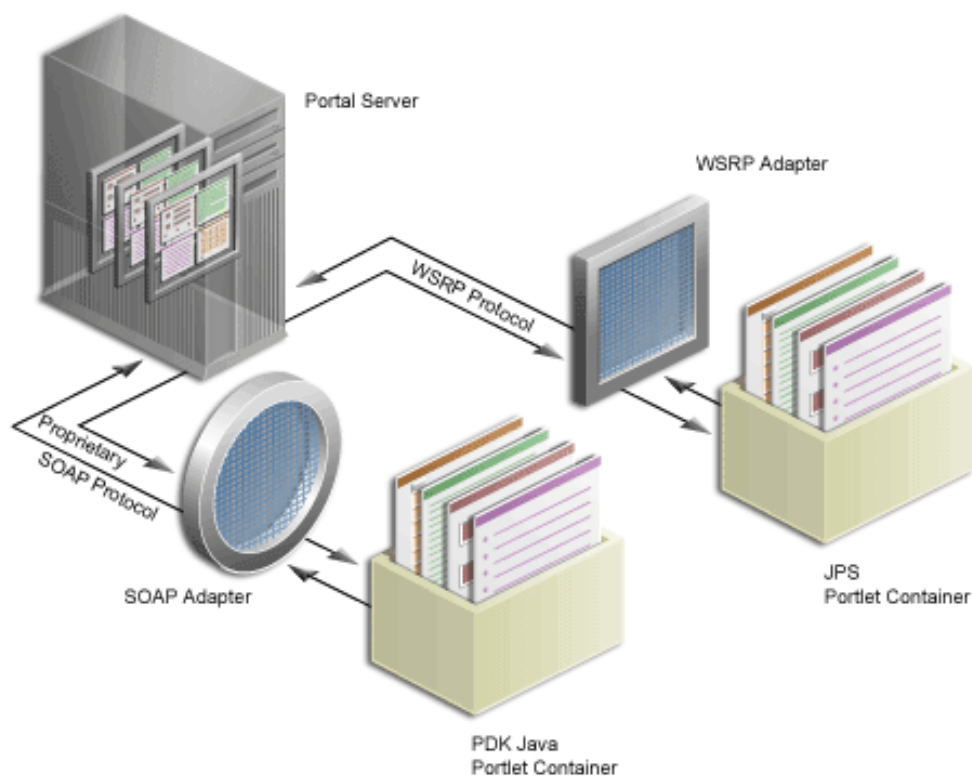
Som udgangspunkt er benyttet samme standard-software til PoC'en, som i borger.dk produktionsmiljøet, nemlig Oracle Application Server 10g (Oracle AS).

For at kunne gennemføre projektet og herunder afprøvning af de valgte standarder, er der installeret yderligere software-komponenter og andre komponenter er blevet opgraderet for at understøtte de valgte standarder i de korrekte versioner. Dette er beskrevet særskilt under afsnittene om hhv. WSRP/JSR168 og IdP.

Borger.dk produktions-miljøet afvikles i et tre-tier server-setup med hhv. database, identity management og portal/web-server på hver sin server, og med separate netværk adskilt af firewalls til hvert tier. Dette viste under opstarten af projektet ikke at være hensigtsmæssigt, hvorfor PoC miljøet er afviklet på to virtuelle servere: en til identity management og database samt en til portal/web-serveren. Begge servere ligger i samme netværkssegment og har været placeret i en DMZ for at minimere problemer med netværksadgang til og fra serverne.

#### **3.1.1 Borger.dk WSRP/JSR168 setup**

Oracle AS understøtter flere portlet-teknologier, herunder Oracles egen PDK Java specification og standarderne WSRP og JSR168 (også kendt som Java Portlet Specification, JPS). JSR understøttes dog ikke direkte, men via "indpakning", hvorefter de afvikles som WSRP portlets, hvor Oracle Portalen både er producer og consumer.



Ved afvikling af WSRP portlets forbindes til en ekstern portlet producer. De to elementer "WSRP Adapter" og "JPS Portlet Container" er altså placeret hos portlet producer (andre steder omtalt provider) og Oracle Portal kommunikerer direkte med den eksterne producer via HTTP.

Afvikling af JSR168 portlets foregår i Oracle AS ved at deploye portlets i en lokal portlet container, med en tilhørende WSRP Adapter. Oracle Portal afvikler dermed JSR168 portlets på samme måde som WSRP portlets, blot med den forskel at portlet producer er placeret lokalt.

#### Benyttet software:

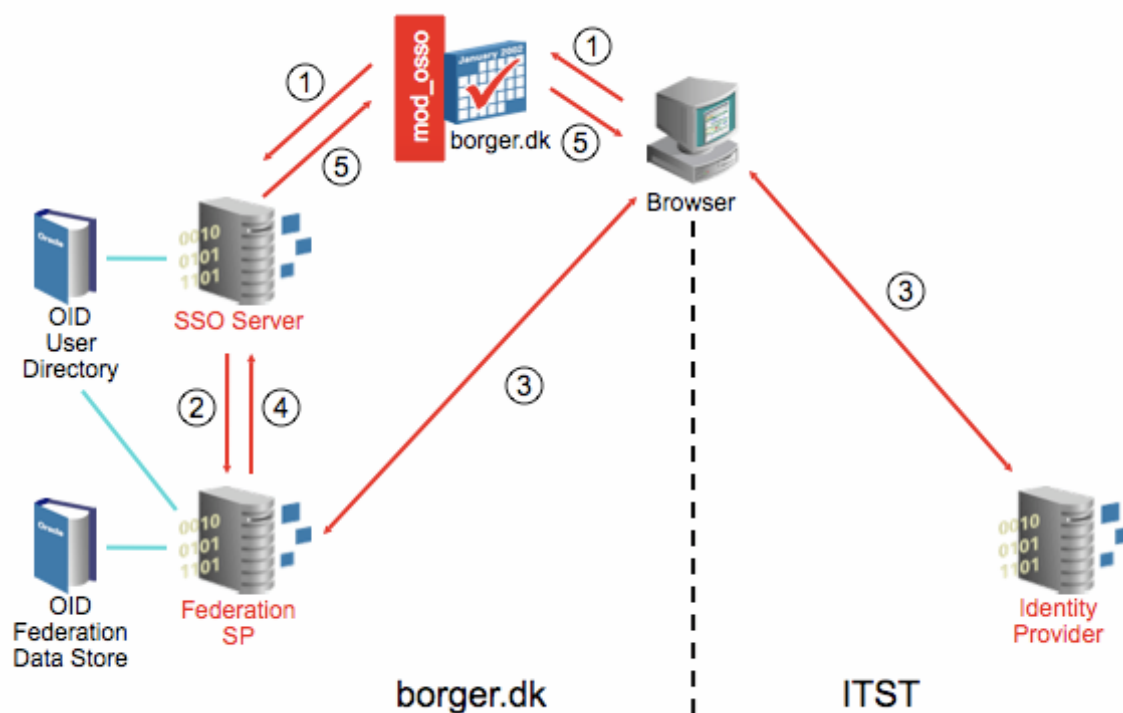
Oracle Application Server 10g version 10.1.2 (som borger.dk):

- Oracle Portal 10g version 10.1.4 (som borger.dk)
- Oracle Containers for Java (OC4J) 10g version 10.1.2 (som borger.dk)
- OracleAS Java Portlet Container version 10.1.4 (ny installation/opgradering til OC4J)

#### 3.1.2 Borger.dk IdP setup

Borger.dk produktions-miljøet benytter sig af Oracle Identity Management til håndtering af autentifikation og administration af brugere og roller. Autentifikation varetages af Oracle Single-Sign-On (SSO) mens brugere og roller administreres af Oracle Internet Directory, der bl.a. består af Oracles LDAP-server.

Oracle Identity Federation (OIF) er Oracles federation server, der bl.a. supporterer SAML 2.0 og kan optræder som både Service Provider (SP) og Identity Provider (IdP). Da OIF ikke benyttes af borger.dk, har det været nødvendigt at installere OIF og i den forbindelse skulle Oracle Identity Management også opgraderes til en nyere version.



1. Brugeren requester en borger.dk side, der er beskyttet af mod\_osso.
2. Oracle SSO er konfigureret til at bruge Oracle Identity Federation (OIF) Service Provider (SP) til autentifikation.
3. OIF SP laver SAML SSO med Identity Provider, der prompter brugeren for brugernavn/password.
4. OIF SP mapper SAML assertion til en lokal bruger og opretter en Oracle SSO session.
5. Der bliver givet adgang til siden af mod\_osso.

#### Benyttet software:

Oracle Application Server 10g version 10.1.2 (som borger.dk):

- Oracle Portal 10g version 10.1.4 (som borger.dk)
- Oracle Identity Management 10g version 10.1.4 (opgraderet fra 10.1.2)
- Oracle Identity Federation 10g version 10.1.4 (ny installation)

## 3.2 Sundhed.dk: BMI selv-service

### 3.2.1 Integrationsform

Integrationsformen der er anvendt er Remote Portlet. Løsningen er baseret på WSRP 1.0 standarden.

### 3.2.2 Case

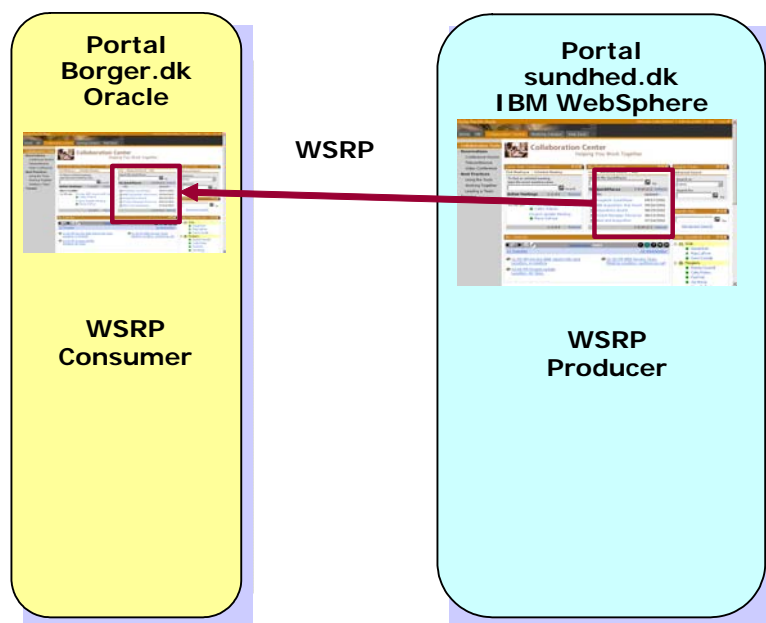
På sundhedsportalen sundhed.dk stilles en række sundhedsrelevante tjenester til rådighed for brugerne gennem en portletbaseret brugersnitflade. Portalen betjener både professionelle indenfor sundhedssektoren og borgere. De sidste kan f.eks. få adgang til oplysninger om indlæggelser og medicinering via recepter.

Til anvendelse i POC blev en selv-service Body Mass Index (BMI) portlet applikation valgt. Denne eksponeres på borger.dk via WSRP protokollen.

### 3.2.3 Overordnet arkitektur

Sundhed.dk afvikles på IBM WebSphere Portal Server. Til POC-afprøvningen blev der opsat en test WebShere Portal Server i et et WM-Ware image, som BMI portletapplikationen blev installeret på.

BMI portlet applikationen trækker ikke på data udefra, men "medbringer" pakket ind i portlet web-applikationen et rammeværk, som anvendes af mange portlets på sundhed.dk.



Som tegningen indikerer er der tale om en punk-til-punkt integration mellem en WSRP web service consumer og en WSRP web service WSRP provider.

Ved opsætning af WSRP forbindelsen sker følgende:

1. På WSRP producer siden – sundhed.dk - konfigureres BMI portletten til at være en WSRP producer. Konfiguration foregår i en administrativ portlet via 2-3 museklik.
2. Herved generes en udefra tilgængelig wsdl hvor de portlets, der er registrerede som producers, indgår (publish).
3. På WSRP consumer siden – borger.dk - konfigureres en WSRP consumer portlet.
4. Link adressen til BMI producer wsdl dokumentet indsættes (find og bind).

**Benyttet software:**

WebSphere Portal Server Version 6

**3.2.4 Bemærkninger**

Der var ingen problemer med WSRP consumer portletten på borger.dk. På BMI portletten viste det sig, at den anvendte et internt objekt i stedet for portlet standard API. Dette gav en fejl under WSRP sessionen. Det blev rettet og herefter kørte portletten via WSRP, som den skulle.

### 3.3 Færdselsstyrelsen: Vognmandstilladelse

#### 3.3.1 Integrationsform

Integrationsformen der er anvendt er Remote Portlet. Løsningen er baseret på WSRP 1.0 standarden.

Denne case er en slags composite application, men ikke som i rapporten "Arkitekturen på en borgerportal" en composite, der baseres på installeret JSR168. Applikationen afvikles hos myndigheden og vises via Remote Portlets.

#### 3.3.2 Case

Færdselsstyrelsen er en composite application, der samler en række myndighedsservices i en sammenhængende proces, hvor man kan søge om en vognmandstilladelse. Det involverer bla. servicekald til CPR/CVR, politiet, skat. For brugeren er der tale om én sammenhængende service.

- Hvert år søger ca. 500 virksomheder om en vejtransporttilladelse til enten gods eller busdrift v.h.a. papir- eller elektroniske ansøgningskemaer
- Op imod 1.100 virksomheder søger ligeledes om fornyelse af deres tilladelse hvert år

#### Udfordring

- Ansøgningsprocessen kræver input fra flere andre offentlige myndigheder, hvilket afføder mere end 7.000 breve og e-mails pr. år med deraf følgende behov for journalisering og sagsbehandling
- Ansøgeren kan indsende ufuldstændige oplysninger, hvilket resulterer i ekstra sagsbehandling og længere sagsgang
- Ansøgeren skal i sagsgangen dokumentere oplysninger, der allerede er i det offentlige besiddelse, men befinder sig i fagsystemerne hos de respektive myndigheder

#### Opgave

- Hvordan kan Færdselsstyrelsen effektivisere denne sagsgang, således at virksomhederne, Færdselsstyrelsen og andre offentlige myndigheder sparer tid og ressourcer?

#### Løsning

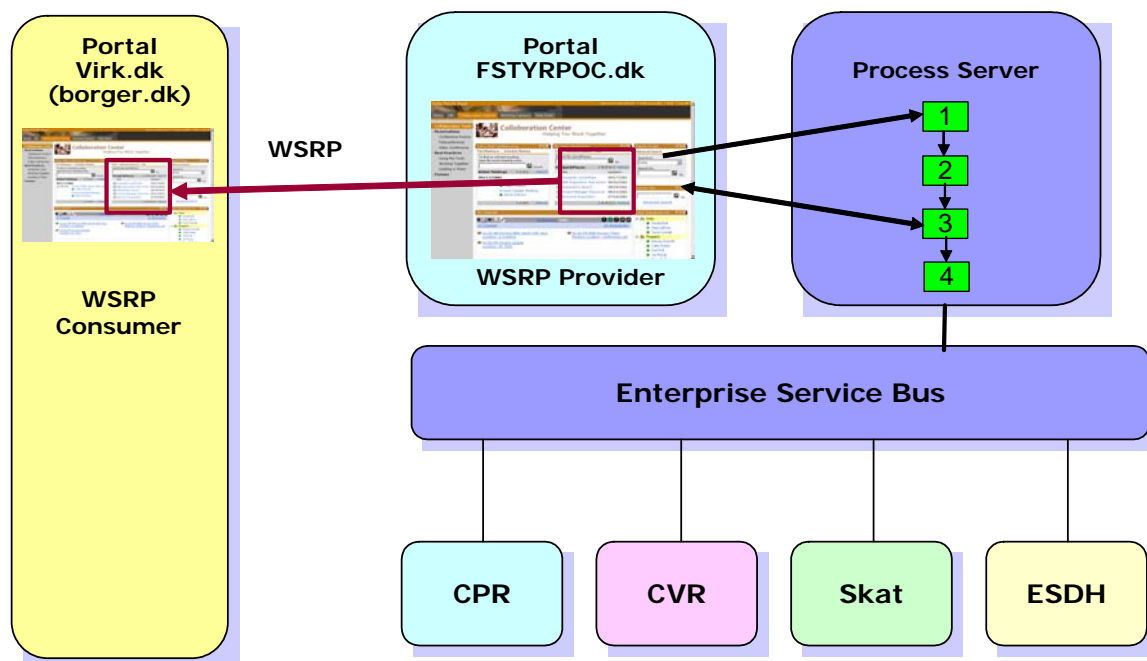
- Ansøgningsprocessen gøres tilgængelig i en portalløsning.
- Her identificeres ansøgeren entydigt v.h.a. sin digitale signatur
- Ansøgeren udfylder de nødvendige informationer – enten fra egen dokumentation (fx. elektronisk revisorerklæring) eller ved at en fælles offentlig procesmotor indhenter oplysningerne fra offentlige registre
- Til sidst sendes den fuldstændige ansøgning til eksempelvis Færdselsstyrelsens ESDH-system, der sender en bekræftelse til ansøgeren, der inkluderer et sagsnummer

### 3.3.3 Overordnet arkitektur

FSTYR POC afvikles på IBM WebSphere Portal Server samt en WebSphere Process Server. Til POC-afprøvningen blev der opsat en test WebShere Portal Server og Process Server i et et WM-Ware image. Vognmands-portletapplikationen blev installeret på Portal serveren og forretningsprocessen (BPEL) blev installeret på Process Server.

Portlet applikationen spiller sammen med BPEL processen på den måde at den dels initierer en BPEL proces instans og dels deltager i nogle af BPEL processens manuelle aktiviteter. Disse opleves af brugeren som en sekvens af skærbilleder i portlet applikationen.

BPEL processen henter oplysninger fra andre offentlige organisationer (CPR, CVR og Skat). Endelig afsluttes BPEL processen med, at der via et webservice-kald oprettes en sag i et ESDH test system (Scanjour Captia).



Ved opsætning af WSRP forbindelsen sker følgende:

1. På WSRP producer siden – FSTYRPOC.dk - konfigureres Vognmandsportlet portletten til at være en WSRP producer.
2. Herved genereres en udefra tilgængelig wsdl hvor de portlets, der er registrerede som producers, indgår (publish).
3. På WSRP consumer siden – borger.dk - konfigureres en WSRP consumer portlet.
4. Link adressen til portlet producer wsdl dokumentet indsættes (find og bind).

Testen er i denne case kompleks, fordi den samlede applikation er distribueret over en række lag, systemer og organisatoriske domæner: To portaler, en proces server og en række eksterne services.

**Benyttet software:**

- WebSphere Portal Server Version 5.1
- WebSphere Process Server Version 6.0
- IBM DB2 UDB Version 8.1
- WebSphere Business Modeler Version 6 (proces modellering, analyse og optimering)
- WebSphere Integration Developer Version 6 (BPEL programmering og service integration)

**3.3.4 Bemærkninger**

Da sikkerhed ikke indgik i POCen var det ikke muligt at identificere ansøgeren. Applikationen har brug for oplysninger om ansøgerens CPR og CVR nummer. Processen der bearbejder og kontrollerer ansøgningen har brug for CPR og CVR numrene for at kalde de underlæggende services fra CPR, CVR og Skat.

I forbindelse med web service sikkerhed på transport niveauet (OWSA Web Service Security Model T), hvor der anvendes SSL til kryptering, kan man evt. vælge at anvende det User objekt der er defineret i WSRP standarden. Dette objekt rummer en række standard brugerinformationer som navn og adresse. For at anvende det, skulle det udvides (er ifølge WSRP tilladt) med CPRNummer og CVRNummer information.

Der var ingen problemer med WSRP consumer portletten på borger.dk.

Service provider applikationen får CVR og CVR nummeret via brugerens login. Hentning af brugerinformation via WSRP web service kaldet kræver derfor en programændring i applikationen. I stedet blev der af hensyn til at kunne gennemføre testen i POCen lavet en stubløsning, som oprettede et test bruger CPR og CVR nummer.

## **3.4 KMD: Mit budget**

### **3.4.1 Integrationsform**

Integrationsformen der er anvendt er Remote Portlet. Løsningen er baseret på WSRP 1.0 standarden.

WSRP dannes fra providersiden ved brug af NetUnity som muliggør at Microsoft .NET løsninger udstilles som Remote Portlets. Denne case er altså et væsentligt supplement til de java-orienterede WSRP afprøvninger for at sikre, at WSRP som standard ikke udelukker den store del af markedet, der er Microsoft-orienteret.

### **3.4.2 Case**

Casen tager udgangspunkt i et løsningen "Mit Budget", som KMD har udviklet og integreret med Ældresagens hjemmeside. (Ældredagen anvender ikke længere produktet). For at kunne integrere med Ældresagens hjemmeside eller for den sags skyld en kommunes hjemmeside anvendes en KMD fælleskomponent "Grafisk udrulning". Denne komponent anvendes af alle KMD's borgerserviceløsninger, så derfor skal det forsøges at kombinere WSRP med denne komponent. Antagelsen er at hvis dette kan lade sig gøre vil det minimere omkostningerne ved at få KMD's borgerserviceløsninger til at "tale" WSRP og dermed at kunne integrere på borger.dk.

Den oprindelige løsning kan nås på vedlagte link

<http://budgetmodul.netborger.dk/budget/default.aspx?p=bs>

For at skifte design ændres p-parameteren til noget, der identificerer et layout. Kan eksempelvis prøves med (p=borgerdk, p=aeldresagen eller p=aarhus).

### **3.4.3 Overordnet arkitektur**

Som det fremgår af case-beskrivelsen er det egentlig ikke "Mit Budget" der skal testes men der imod komponenten "Grafisk Udrulning". "Mit Budget" er valgt da den ikke har bindinger til fagsystemer hvilket bare ville medføre besvær i en testsituation. Som back-end er der dermed kun en MS SQL database som tilgås med ODBC.

Setup'et ser således ud.

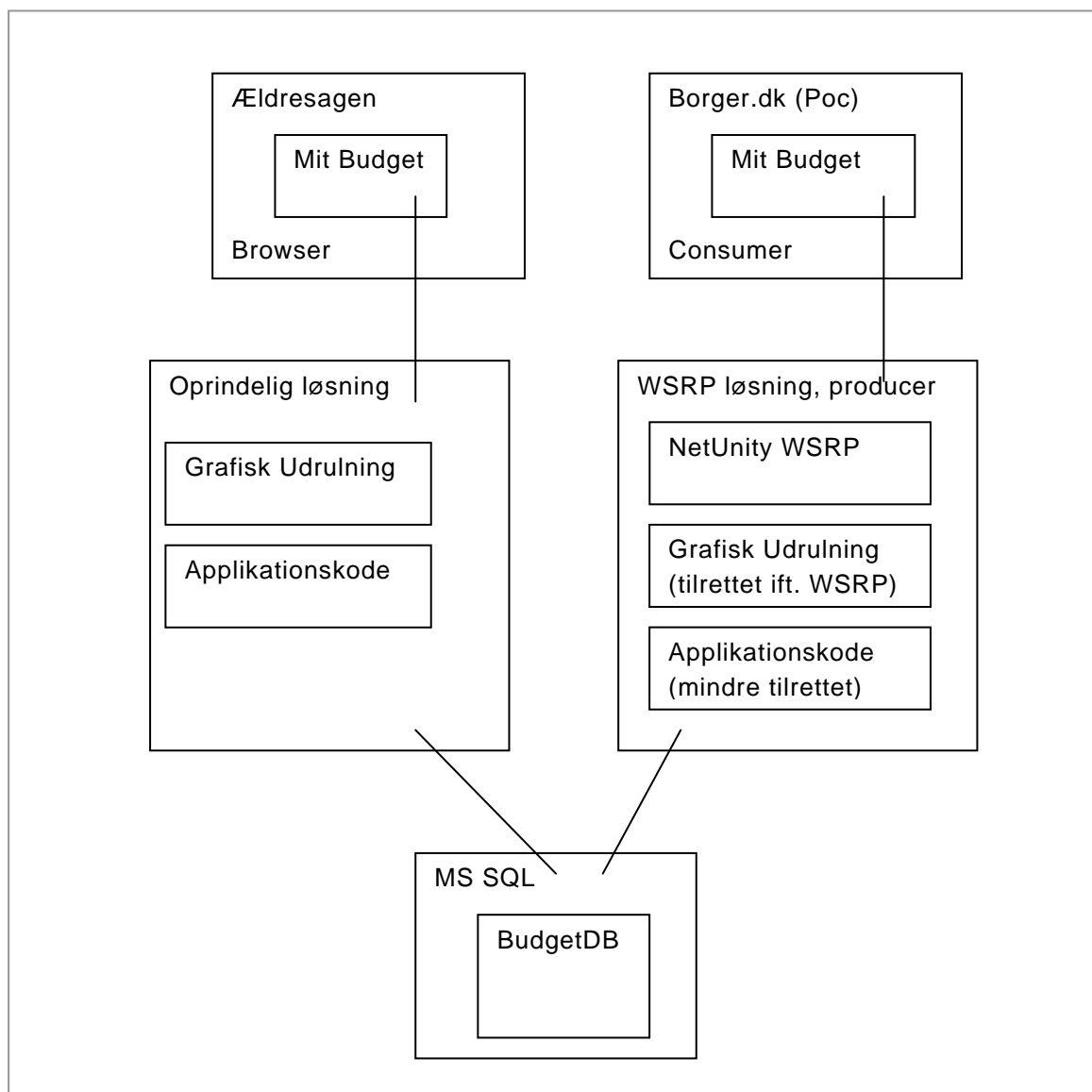


Diagram over den oprindelige løsning og WSRP løsningen

WSRP løsningen udstiller en provider på URL'en

<http://minside.test.dk/wsrp/producer1.asmx?Operation=WSDL>

Servicen er placeret hos KMD og den tilgås via SOAP/HTTP.

## **3.5 IBM: CVR service**

### **3.5.1 Integrationsform**

Integrationsformen der er anvendt er Portlets. Løsningen er baseret på JSR168 Portlet.

Der er forsøgt afprøvet en composite application, men erfaringer er primært baseret mere simple applikationer.

### **3.5.2 Case**

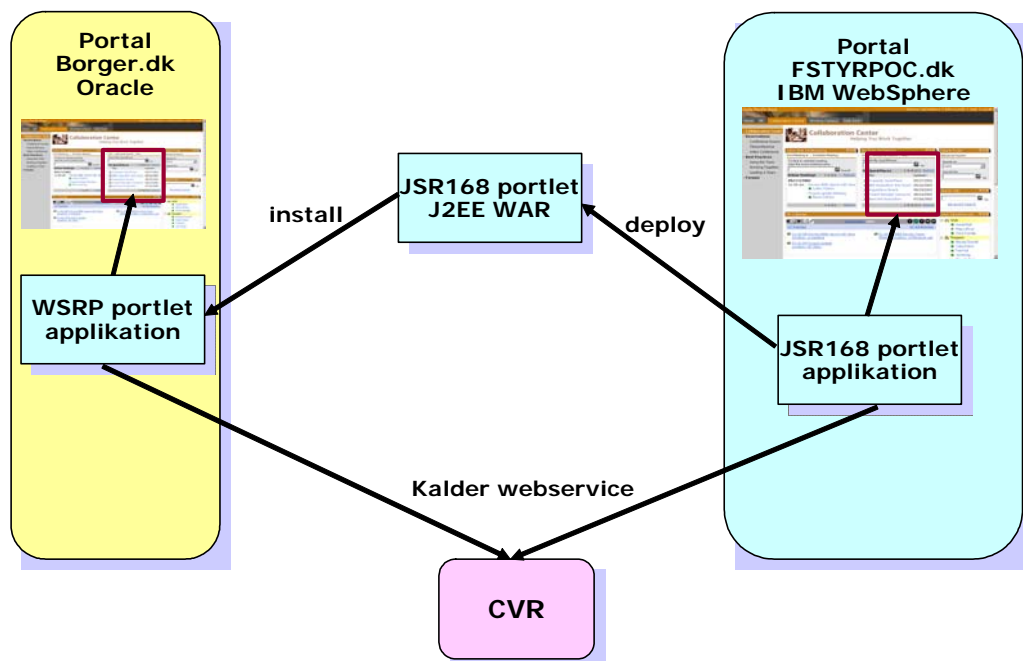
Der var oprindeligt tænkt en *composite application* ved brug af JSR udgave af Færdselsstyrelsen. Det var ikke muligt, da der er anvendt portal-teknologi udover JSR168 standarden i denne implementering.

Derfor blev der i stedet installeret og afprøvet en alternativ portlet, som henter oplysninger om en virksomhed. Portlet applikation er en lille hjælpeservice, hvor man kan søge efter oplysninger om en virksomhed ud fra dens CVR nummer. Det sker med et vist genbrug af løsningsarkitekturen fra Færdselsstyrelsen case'en.

Erfaringer er desuden baseret på en helt simpel JSR168 installation, hvor deployment udfordringer og andet gjorde det umuligt at foretage fuld afprøvning og drage konklusioner på de øvrige nævnte services.

### **3.5.3 Overordnet arkitektur**

Der er udviklet en Portlet efter JSR168 standarden på IBMs Web Sphere platform. Denne portlet er herefter deployet til Oracle Portal, hvor koden afvikles.



Der er altså tale om afvikling på samme server som portalen og ikke som ved brug af WSRP en remote portlet i et distribueret miljø.

Ved overførsel af en JSR168 portlet sker følgende:

- JSR168 portlet applikationen blev lavet med IBM WebSphere Portal Factory ud fra cvr.wSDL dokumentet, udstillet af Krak.dk
- Portlet Factory samler portlet applikationen i en arkivfil i henhold til J2EE standardens regler for pakning af web applikationer i web archive files (war).
- Portlet war filen blev deployed, installeret og testet på WebSphere Portal server
- Herfter blev portlet war filen sendt til borger.dk udviklingsmiljøet med e-mail.
- Portlet war filen blev herefter lagt in på Oracle Portal Server
- JSR168 portlet applikationen afvikles som en WSRP portlet, da Oracle portalen foretager indpakning og optræder som både producer og consumer (JSR168 understøttes ikke direkte).

#### Benyttet software:

- IBM WebSphere Portal Server Version 6
- IBM WebSphere Portal Factory Version 6 (udvikling af portlet)

#### 3.5.4 Bemærkninger

Det viste sig af Oracle Portal ikke har sin egen JSR168 container. Man er derfor nødt til at pakke den ind som en WSRP provider, før den kan anvendes på Oracle Portal, med den øgede kompleksitet og det performance overhead, som dette medfører.

Testen fejlede fordi WebSphere Portal bruger en open source xerces xml parser fra Apache mens Oracle Portal anvender sin egen xml parser. Begge platforme understøtter XML

og Web Services, som de skal for at overholde J2EE standarden, men Oracle og IBM har valgte forskellige implementeringer af deres XML og Web Service runtime miljøer.

Løsningen på dette er at tage eventuelle runtime-platform forskelle med i portlet WAR arket. I dette tilfælde xerces biblioteket fra Apache. Dette kan lade sig gøre, men gør dels web applikationen større, øger kompleksiteten og introducerer nye fejlmuligheder.

## 3.6 Rødovre Kommune: ” e-posthus.dk”

### 3.6.1 Integrationsform

Integrationsformen der er anvendt er Link og iFrame m/Single Sign-on (SSO).

Link og iFrame er under PoC afprøvet samlet, da det teknisk set har de samme udfordringer. Set med informationsarkitekturøjne kan disse være meget forskellige integrationsformer, men det giver ikke mening at afprøve og erfaringsopsamle som om disse var vidt forskellige udfordringer set i forhold til de standarder, der skal afprøves på tværs af tekniske platforme.

### 3.6.2 Case

e-posthus.dk er en post og kalender portalløsning hos Rødovre Kommune, der med tiden kommer til at favne en række services - såvel kommunale som andre.

Det bør være således at borgeren ikke møder flere logins, når han/hun bevæger sig mellem en Borgerportal og Rødovres e-posthus.dk løsning - uanset hvilken portal denne vælger (først).

Rødovre ønsker en løsning som her og nu giver single sign-on til kommunens interne tjenester, og kan anvendes med eksisterende login metoder som brugernavn/kodeord, borgers OCES certifikat, og samtidig være forberedt til den fremtidige fælles offentlige login service med anvendelse af SAML 2 tokens.

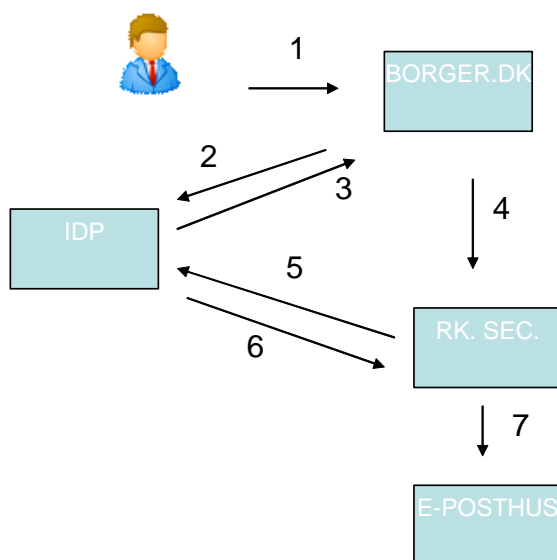
Der blev således testet en række scenarier med kommunens Epost løsning som eksempel på en service provider:

1. Kommunal single sign-on med brugernavn og kodeord til kommunens e-posthus.dk.
2. Kommunal singlesign-on med borgerens OCES certifikat til kommunens e-posthus.dk.
3. Fælles offentlig single sign-on med OCES certifikat via den fælles offentlige login-service (IDProvider) til kommunens e-posthus.dk med link videre til borger.dk
4. Fælles offentlig single sign-on med OCES certifikat via den fælles offentlige login-service (IDProvider) til borger.dk med link videre til kommunens e-posthus.dk

SSO scenarie 4 - detaljeret:

1. Borgeren kommer til Borgerportalen  
Borgeren anvender en ressource/service, der kræver login
2. Login rettes mod IDProvideren
3. ID Provideren returnerer et SAML token til login på Portalen  
Borgeren vises den ønskede (adgangsbegrænsede) side på Borgerportalen
4. Borgeren klikker på link til Rødovres e-posthus.dk

5. Sikkerhedskomponenten (TFIM) checker SAML token op mod IDProvideren
6. Accepten returneres fra IDProvideren
7. Brugeren får adgang til Rødovres e-posthus.dk



I virkeligheden var der ikke den store forskel på de to fælles offentlige single sign-on scenarier (3 og 4) da borger.dk og Rødovre site principielt har den samme rolle som service provider. Links begge veje lykkedes da af samme grund også.

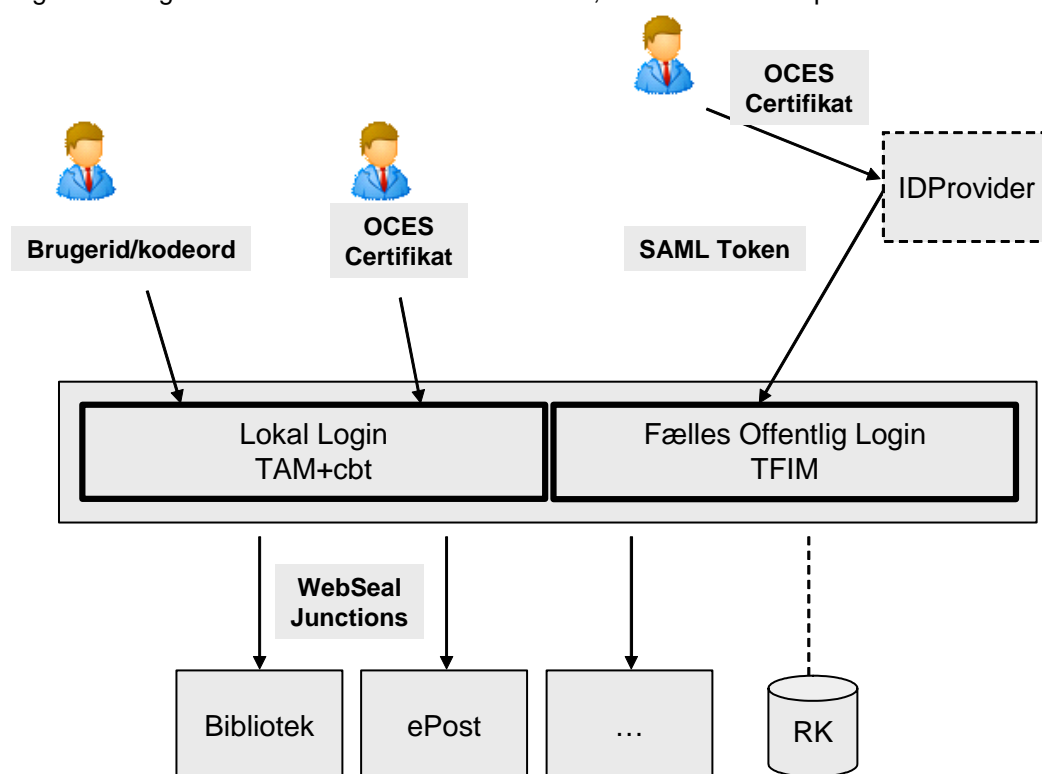
### 3.6.3 Overordnet arkitektur

I de to scenarier med fællesoffentlig single sign-on indgik således tre hovedaktøren:

- IdProvideren, som i POC var en Ping Identity installation hos ITST
- Borger.dk portalen og dens sikkerhedskomponenter til håndtering af SAML login
- Rødovres testinstallation med e-posthuses og kommunens sikkerhedsløsning til håndtering af de ovennævnte login metoder (lokal login med brugernavn/kodeord og OCES certifikat samt fællesoffentlig login med SAML token).

## Arkitekturen i Rødovre Kommune:

Det følgende diagram viser sikkerhedsarkitekturen, som den blev afprøvet i denne PoC:



De kommunale services er illustreret med Epost og Bibliotek. ePost indgik i afprøvningen. e-posthus.dk er en Domino Mail Server, som tilgås via http med Domino Web Access.

RK er et LDAP brugerkatalog over Kommunens borgere og personer, som kan tilgå kommunens systemer.

Scenarie 1 og 2 med lokalt login med single sign-on og support for Brugerid/kodeord og OCES certifikat understøttes af IBM Tivoli Access Manager(TAM) med IBM cbt til understøttelse af OCES certifikathåndteringen. I TAM konfigurerer man såkaldte "junctions, som sørger for at brugerens login transformeres videre på en for de bagvedliggende systemer forståelig måde.

I scenarie 3 og 4 tilgås de kommunale systemer via en fælles offentlig login service (illustreret med IDProvider) er det IBM Tivoli Federated Identity Management (TFIM), der kan modtage det afsendte SAML-token og sammen med TAM omsætte det gennem en junction til et forståeligt login for de bagvedliggende systemer som f.eks. ePost.

## Anvendt Software:

- IBM Tivoli Federated Identity Management Version 6.1
- IBM Tivoli Access Manager Version 6.0 inklusiv
  - IBM DB2
  - Tivoli Directory Server

- IBM cbt (ændring af OCES certifikater)

#### **3.6.4 Bemærkninger**

Det lykkedes at konfigurere og teste alle de fire nævnte scenarier med succes.

## 4 Afprøvning og erfaringer

### 4.1 Generelle erfaringer og anbefalinger

#### Standarder og produktunderstøttelse

Der findes veldefinerede standarder til at foretage udvikling ud fra. Det være sig JSR168 til portlets, WSRP1.0 til Remote Portlets og SAML til Single Sign-on i linking scenarier. Der findes desuden produktunderstøttelse af disse standarder på nuværende tidspunkt.

Dog er det således, at man typisk skal anvende en af de nyere versioner af leverandørernes produkter, for at arbejde med disse standarder (se mere under hver enkelt integrationsform og vær opmærksom på at det vil være forskelle fra leverandør til leverandør). Samtidigt skal man være opmærksom på, at der ikke er mange i Danmark, der har erfaringer med at arbejde med disse standarder.

#### Udvikling i et distribueret miljø

Selv hvor standarder er på plads og hvor interoperabilitet umiddelbart ikke giver udfordringer er der altid udfordringer i et distribueret miljø.



Vær opmærksom på, at debugging i et distribueret miljø ikke er trivielt. Det er vanskeligere at finde fejl og ofte en god ide at afsætte tid til at "begge sider" af en integration sammen prøver at finde problemer og løsninger.


Man kan evt. lave en samlet testinstallation eller fysisk sidder sammen, hvis man kan få den nødvendige netværksadgang til serverne, der indgår i integrationen (minimum test og gerne udvikling og test).

Særligt i forhold til test er det en udfordring at hver myndighed ikke kan forventes at have adgang til samme produkter og produktversioner som de skal integrere op mod. Selv om standarden ligger fast kan det have interesse at et bestemt produkt i rette version og med en bestemt konfiguration er til rådighed. Det var været en udfordring gennem hele PoC projektet at sikre gensidig adgang til test. Bemærk at der er tale om miljøer, der ikke er at sidestille med alm. testmiljøer, da der er behov for "adgang udefra".




Der vil oftest blive behov for yderligere et testmiljø udover den klassiske udvikling, test, produktion. Dette testmiljø har til formål at kunne teste integrationer med portalen eller andre myndigheder uden at blokere for udviklingsteamets rådighed over udviklings- og testmiljøet.

Det må forventes at der implementeres formelle afprøvningsprocedurer, der skal gennemføres før nye services kommer på portalerne. Der kan dog være behov for mere uformel test undervejs i serviceproviders udviklingsforløb. Flere deltagere i PoC har udtrykt ønske om testmiljøer, der kontinuerligt er til rådighed – også lang tid før lancering af nye services.

	<p>Der er et ønske blandt flere myndigheder og deres tekniske leverandører om, at der stilles et særligt testmiljø til rådighed, som kan anvendes <i>under udviklingsfasen</i>.</p> <p>Det skulle være en slags "sandkasse", som man kan prøve nogle ting af i for at se, hvordan det vil virke på portalen – før man når til en mere formel test koordineret med consumer -siden.</p> <p>Man kan evt. forestille sig et miljø, man kan søge om adgang til og som automatisk rulles tilbage til en bestemt tilstand med nogle dages interval.</p>
---	---


### Design retningslinier og visuel integration

En række integrationsformer giver i sig selv nogle begrænsninger og muligheder i forhold til at styre layout og kan i mere eller mindre grad understøtte en visuel integration. Der er dog nogle generelle overvejelser der går igen – uanset integrationsform.

	<p>Det anbefales, at man etablerer en proces, hvor en ny service skal gennem en række compliance test, før den introduceres på portalen.</p> <p>Som minimum bør følgende overvejes:</p> <ul style="list-style-type: none"> <li>• W3C check for HTML (herunder valg mellem HTML 4.01 og XHTML)</li> <li>• Check for korrekt brug af pre-fixes og class names i JSR168 og WSRP</li> <li>• Manuelt check for GUI compliance</li> </ul>
--	---

Eksempelvis er der behov for enighed om hvilke HTML version, der tillades lige som man skal have procedurer, der sikrer at W3C standarderne overholdes af alle – hvis brugeren skal stå med en brugeroplevelse af, at portalen samlet set overholder disse standarder. På samme måde kan der for bestemte integrationsformer være behov for yderligere compliance test (specielt portlets - JSR/WSRP).

I sidste ende er det design som serviceprovider implementerer det mest afgørende for den endelige brugeroplevelse. Derfor er der behov for, at services, der ønskes tæt integreret til portalen, overholder nogle designretningslinier og ikke mindst er designet med tanke på at skulle vises på en portal (kan vises med begrænset plads og gerne være fleksibel).

	<p>Uanset hvad integrationsformen måtte tilbyde, er der en række helt generelle designvalg, som stadig har afgørende betydning for, hvordan en service tager sig ud overfor slutbrugeren!</p> <p>Eksempelvis er det primært nogle designvalg foretaget af udvikleren/designeren hos serviceprovider, der afgør om løsningen kun tager sig pænt ud, når der afsættes et bestemt antal pixels på skærmen.</p>
---	---

## 4.2 Link eller iFrame m/Single Sing-on (SSO)

Integrationsformen link m/Single Sign-on handler basalt set ikke om at kunne link'e, da der absolut intet nyt og uafprøvet er i denne integrationsform. Det er basal HTML og link/anchor (<A href="...">link text</A>). Udfordringen ligger i at man overfører information om, at brugeren er logget ind én gang - og attributter, der fortæller, hvem brugeren er. Det sker med brug af SAML 2.0 standarden.

Links og iFrame giver teknisk set samme udfordringer og er derfor behandlet under ét.

Scenariet er følgende:

1. Borgeren kommer til én portal
2. Borgeren anvender en ressource/service, der kræver login
3. Login sker via *fælles logintjeneste* – en såkaldt Identity Provider (IdP)
4. Borgeren vises den ønskede (adgangsbegrænsede) side på portalen
5. Borgeren klikker på link til en anden portal (hos anden organisation)
6. Siden på den "anden portal" vises uden yderligere login

### 4.2.1 Integrationsformen og standarder

Link m/SSO er baseret på standarden SAML 2.0. SAML 2.0 er en "approved" OASIS standard og en "anbefalet" standard i OIO kataloget.

Der er den udfordringer i forhold til valg af standarder, at profilering af SAML i forhold til browser SSO er relativ klar, mens der ikke er en SAML profil for, hvordan SSO etableres på Portlet og web service baserede services og hele vejen ned til back end systemer hos myndigheden.

PoC har kun afprøvet SAML 2.0 i forhold til browser SSO scenarier.

### 4.2.2 Produktunderstøttelse og interoperabilitet


Der findes i dag produktunderstøttelse for standarden og både consumer og serviceprovider siden af integrationen er baseret på standardprodukter fra deres leverandører (hhv. Oracle og IBM). Bemærk at Microsoft endnu ikke understøtter SAML 2.0 (kun SAML 1.0) men forventes at understøtte SAML 2.0 *tokens* fremover.

SAML 2.0 og uddelegeret SSO er ikke noget portalprodukterne understøtter "out of the box". Der kræves installation af ekstra komponenter eller ligefrem selvstændige produkter. Disse produkter kræver typisk licenser, der medfører omkostninger af ikke uvæsentligt størrelse<sup>1</sup>.

---

<sup>1</sup> Der findes open source implementeringer, men disse er ikke afprøvet i PoC sammenhæng.

Yderligere var erfaringen i PoC forløbet, at det var nødvendigt med en række opgraderinger og patches på Oracle portalen, som ligger til grund for Borger.dk, for at forskellige versioner spillede korrekt sammen. Det vil være et typisk forløb, der kan forventes, fordi der er tale om relative nye standarder og ny teknologi.

	Du skal ikke som myndighed forvente, at alle leverandører og deres danske implementeringspartnere har SAML 2.0 erfaringer i 2007-2008.
---	--

Generelt spiller SSO baseret på SAML fint på tværs af platforme. SAML er en XML standard som ikke kræver brug af bestemte platforme eller portalløsninger.

Der er dog konstateret mindre udfordringer med interoperabilitet i forbindelse med etablering af det trustede miljø mellem Identity Provider og serviceproviders. Udveksling og indlæsning af XML konfigurations-filer var ikke helt så smertefri, som man kunne håbe, men det var dog absolut muligt at opsætte et trusted miljø mellem produkter fra forskellige leverandører.

Der har ligeledes været udfordringer med visse browser versioner (versioner af Internet Explorer) undervejs i afprøvningen. Der er ingen klare konklusioner, men det forventes ikke at have direkte relation til de afprøvede standarder og produkter og dermed være hindrende for at opbygge en stabil løsning baseret herpå.

Samlet set efterlades der et indtryk af standarder og teknologi, der kan bringes til at virke og tale sammen på tværs af leverandører allerede i dag. Man behøver således ikke vente på produktunderstøttelsen. Man kan dog forvente flere udfordringer i dag end med mere modne standarder og etablerede produkter. Det må forventes, at der sker en læring efterhånden som flere og flere bevæger sig over på disse teknologier, hvorfor man kan forvente kortere tid til udvikling og integration som tiden går i perioden 2008-2010.

#### **4.2.3 Mulighed for styring af brugergrænsefladen**

Grundlæggende er der tale om at man oplever en helt ny struktur, ny navigation, nyt layout mv. med tydelig afsenderskift, når man anvender integrationsformen link m/SSO. Det er dog noget man kan forsøge at omgå med forskellige teknikker.

Teknisk set er det muligt at anvende SSO i forbindelse med såvel links som iFrames. Det er dermed et forretningsmæssigt valg om man ønsker at "link'e ud af portalen" eller indlejre HTML sider i en portalside via iFrame.

På samme måde skal der træffes et valg i forhold til, om man ønsker at tilpasse link'et og frame'et indhold mest muligt til portalen og give et sammenhængende visuelt udtryk *eller* om man faktisk ønsker at give et tydeligt signal om, at der er tale om en ny afsender og at brugeren befinder sig i en anden kontekst end selve portalen.

Når man link'er vil man typisk åbne et nyt vindue og det er – som nævnt ovenfor - et politisk valg, om man ønsker at lade siden åbne "as is" eller om man vil forsøge at efterligne portalens udseende. Det samme gør sig i princippet gældende med iFrame, men her vil det være mere oplagt, at man anvender en teknik til at tilpasse layout, da man mht. navigation mv. fortsat er i portalens miljø (men dog har indlæst en HTML side fra en serviceprovider og viser denne indlejret).



Vær opmærksom på, at når iFrames tilpasses portalens layout, sker det på baggrund af *en aftale med serviceprovider*, som tilpasser sit layout til portalen (evt. med mulighed for flere forskellige layouts til flere portaler/anvendelser).

Hvis man ønsker at give en ensartet visuel oplevelse, kan man benytte en teknik, hvor serviceprovider på baggrund af hvem referer (den der linkes fra) er pålægger en bestemt style, således at farver, skrifttyper mv. tilpasses og ligner den portal man er linket fra. Det er absolut teknisk muligt og er en teknik, der blandt andet anvendes i KMDs webløsninger på det nuværende borger.dk, jf. evt. KMD MitBudget case'en:

<http://budgetmodul.netborger.dk/budget/default.aspx?p=bs>

<http://budgetmodul.netborger.dk/budget/default.aspx?p=aarhus>

En lignende teknik findes i standardportaler som IBMs WebSphere Portal, hvor de temaer og præsentationstemplates, der anvendes, kan styres af hvor http forespørgslen kommer fra (en bestemt kommune, borger.dk eller en anden kendt portal), hvilken browser (eller mobil enhed) osv. PoC har ikke forsøgt at afdække andre portalprodukters (Oracle, BEA, Microsoft mv.) tilbud på dette område.

Det betyder at kontrol over style ligger hos serviceprovider og at denne dels skal designe efter at kunne anvende forskellige stylesheets og dels skal vedligeholde et antal forskellige stylesheets (ét for hver consumer der måtte have ønske om selv at kunne bestemme over det grafiske snit).



Det anbefales, at man etablerer et regelsæt for, hvad man tillader i iFrames. Det er eksempelvis teknisk muligt for en serviceprovider, at angive at dennes service skal "overtage" browser vinduet. Det er ikke nødvendigvis noget, man bør tillade.


Det er uproblematisk at lade portalen styre, hvor meget plads der afsættes til iFrames. Resultatet kan dog blive en mindre god brugeroplevelse, hvor der kommer scroll bars i begge retninger. Det valgte design er derfor – som altid – væsentligt for slutbrugeroplevelsen.

#### 4.2.4 Mulighed for interaktion med andre services


Når man anvender links bevæger brugeren sig over i en helt ny kontekst og det giver derfor ikke mening at diskutere samspillet med andre services på portalen.

iFrames giver umiddelbart ingen mulighed for inter-frame eller frame-portlet kommunikation og giver dermed de samme begrænsninger som WSRP 1.0. Work-arounds kan tænkes men er ikke forsøgt i PoC.

Når en iFrame udgør en del af en portalside, vil handlinger i andre services typisk betyde, at iFrame'en genindlæses og returnerer til startside. Resultatet vil afhænge af, hvordan løsningen er opbygget hos serviceprovider, men simpel navigation i iFrame-vinduet huskes ikke.

	Hvis du designer en løsning til frameing i et vindue af begrænset størrelse, som skal vises sammen med andet indhold, bør tilstand håndteres, så kald af andre services ikke giver utilsigtede resultater i frame vinduet (f.eks. retur til startside).
---	---

Udover ovennævnte udfordring kan iFrames ganske let anvendes til at sammensætte indhold fra flere leverandører.

	iFrames giver umiddelbart ingen begrænsninger i forhold til formulærnavngivning, brug af javascript og herunder eksempelvis pop-ups, da der er tale om selvstændige dokumenter set fra browseren.
--	---


Denne integrationsform giver ingen begrænsninger i forhold til kald af bagvedliggende services. Eftersom servicen fortsat afvikles hos serviceprovider som altid, er der fuld adgang til web service kald, brug af back end legacy og læsning fra databaser mv.

Integrationsformen har i denne sammenhæng den fordel at myndigheden som er serviceprovider, har sikkerhed for hvem slutbrugeren er og dermed kan stille følsomme data fra back end systemer til rådighed uden at skulle indgå særlige aftaler om trust med portalejeren (udover hvad SSO setup'et med en fælles logintjeneste kræver).


#### **4.2.5 Performance og driftsmæssig tilgængelighed**

Performance ved links er som udgangspunkt uændret, da drift sker hos serviceprovider på samme vis som før integrationen. Der vil være et mindre overhead i forbindelse med redirects og check af autentificering i SSO scenariet, men det opvejes så rigeligt af, at brugeren sparer tiden til selv at logge ind. Det samme gør sig gældende ved iFrame ved den lille tilføjelse at svartiden set med brugerøjne naturligvis afhænger af, at begge servere (portal der frame'er og serviceprovider med selve servicen) svarer hurtigt.

Såvel link som iFrame har et klart snit i forhold til hvor kode afvikles og hvem der har det driftsmæssige ansvar. Der sker ikke implementering på portalen, som kan give anledning til et uklart billede af, hvor ansvaret ligger i tilfælde af fejl og nedbrud.

	<p>Det anbefales, at man i fælles brugerstyringsammenhæng får etableret fælles retningslinier for timeout og Single log-out.</p> <p>Som det er i dag kan hver service have sin egen timeout og hver især forholde sig til om man skal logge ind forfra på logintjenesten (forced login) eller blot ("silent") have verificeret, at brugeren stadig er autentificeret her. Det kan give en utilsigtet forvirrende brugeroplevelse med forskellige services, der timer ud hver for sig (og håndterer dette forskelligt).</p>
---	--

Der vil i SSO tjenester opstå en væsentlig driftsmæssig afhængighed af logintjenesten (Identity provider). Når logintjenesten er nede vil integrationsformen resultere i synlige fejl for brugeren. Man kan dog rent teknisk forestille sig flere alternative logintjenester, der giver adgang til samme service hos myndigheden og evt. portalerne. Om det i praksis bliver tilfældet er et politisk valg.

	<p>Det anbefales at der etableres et slags "katalog" over links. Der er behov for at holde styr på om links fortsat er tilgængelige.</p> <p>Man kan overveje om der kun linkes indirekte (en slags link broker).</p>
---	--

#### 4.2.6 Opsummering: Fordele og ulemper ved denne integrationsform

Fordele +	Ulemper -
<ul style="list-style-type: none"> <li>• Links og iFrame er en oplagt og simpel måde at genbruge eksisterende web løsninger uden meget omskrivning (om nogen overhovedet)</li> <li>• Serviceprovider pålægges umiddelbart ingen restriktioner i brugerdialogen og kan råde over hele browservinduet ved links (og næsten ved en stor frame)</li> <li>• Platformsuafhængighed. Der linkes jo bare mellem løsninger</li> <li>• Der er et klart snit mellem consumer og serviceprovider i forhold til ansvar for udvikling og drift</li> <li>• Et klart scenarie for hvordan myndigheden har sikkerhed for, hvem følsomme data udleveres til (slutbrugeren)</li> </ul>	<ul style="list-style-type: none"> <li>• Services optræder ikke som integrerede på portalen (jf. ønske om borgernes fælles indgang)</li> <li>• Styling kræver work arounds og koordinering med serviceprovider</li> <li>• SAML 2.0 og brug af Identity provider er ikke velafprøvet teknologi (få erfaringer med produkterne og meget få i Danmark har i praksis arbejdet med standarderne)</li> <li>• Microsoft understøtter endnu ikke SAML 2.0</li> <li>• SAML og fælles logintjenester kræver typisk anskaffelse af nye produkter som påfører serviceprovider nogle omkostninger</li> </ul>

### 4.3 Remote Portlet – WSRP

Remote Portlets giver mulighed for at portlets vises ét sted, men udvikles og driftes et andet (deraf "remote"). Integrationen er underliggende baseret på Web service kald.

#### 4.3.1 Integrationsformen og standarder

Remote Portlets er baseret på WSRP 1.0 standarden. WSRP har været en approved standard siden 2003. WSRP 2.0 er på vej, men er endnu ikke klar og der er ingen produktunderstøttelse. Remote Portlets bør derfor for nuværende baseres på WSRP version 1.0.

Der er ingen profil for SAML understøttelse sammen med WSRP og integrationsformen er afprøvet uden sikkerhed. Der kan konstrueres taktiske sikkerhedsløsninger med brug af kendt teknologi for Web Service sikkerhed – eksempelvis med brug af SSL (som OWSA Model T). Dette er dog ikke afprøvet.

#### 4.3.2 Produktunderstøttelse og interoperabilitet

WSRP er understøttet af en række leverandører. I PoC er dette afprøvet på IBM WebSphere portalen og Oracle Portal samt på Microsoft .NET platformen med brug af en 3. parts komponent (NetUnity).

Produktunderstøttelsen er således tilstede i dag, men man skal forvente at det i nogle tilfælde kan være nødvendigt at opgraderer eksisterende løsninger, som måtte kører på ældre portalplatforme.

	<p>Understøttelse af WSRP <i>kan</i> kræve opdatering af den eksisterende portal platform hos serviceprovider.</p> <p>Standarden er dog fra 2003, så selvom WSRP ikke har den store udbredelse i dag kan din platform meget vel understøtte Remote Portlets allerede.</p>
---	---

Interoperabilitet mellem de forskellige leverandører har ikke givet mange problemer. Der er konstateret forskelle i tolkning Window state (nærmere bestemt "maximize") mellem IBM og Oracle, således at den ene portal viser en portlet med maksimal bredde (IBM) mens den anden åbner et nyt vindue (Oracle).

Der er desuden lidt forskel og ikke umiddelbart fuld understøttelse (default styles der kan vælges og konfigureres med peg-og-klik) af de portlet-relaterede HTML classes som WSRP specifikationen angiver (til style brug). Muligheden for styling er dog understøttet og overholdelse af WSRP specifikationen er det rigtige udgangspunkt for at opnå interoperabilitet. Der vil dog blive behov for at udvide aftale om classes og styles ud over hvad specifikationen håndterer.



De største udfordringer med WSRP, når der udvikles services hos en myndighed og integreres med en anden stammer ikke fra interoperabilitet knyttet til standarden, men til de generelle udfordringer der er med at arbejde i et distribueret miljø.

Det giver særlige udfordringer i forhold til test og fejlfinding og det stiller krav til de miljøer, der anvendes – og adgangen hertil.

Microsoft kan consume WSRP portlets i Sharepoint, men dette er ikke afprøvet. Microsoft har ikke selv en provider understøttelse i deres produkter og WSRP må fortsat forventes at være lettest med en java platform. NetUnity komponenten har dog vist at det *er* muligt via 3. parts komponenter at udstille .NET løsninger som WSRP uden nogen væsentlige interoperabilitets problemer.

#### 4.3.3 Mulighed for styring af brugergrænsefladen

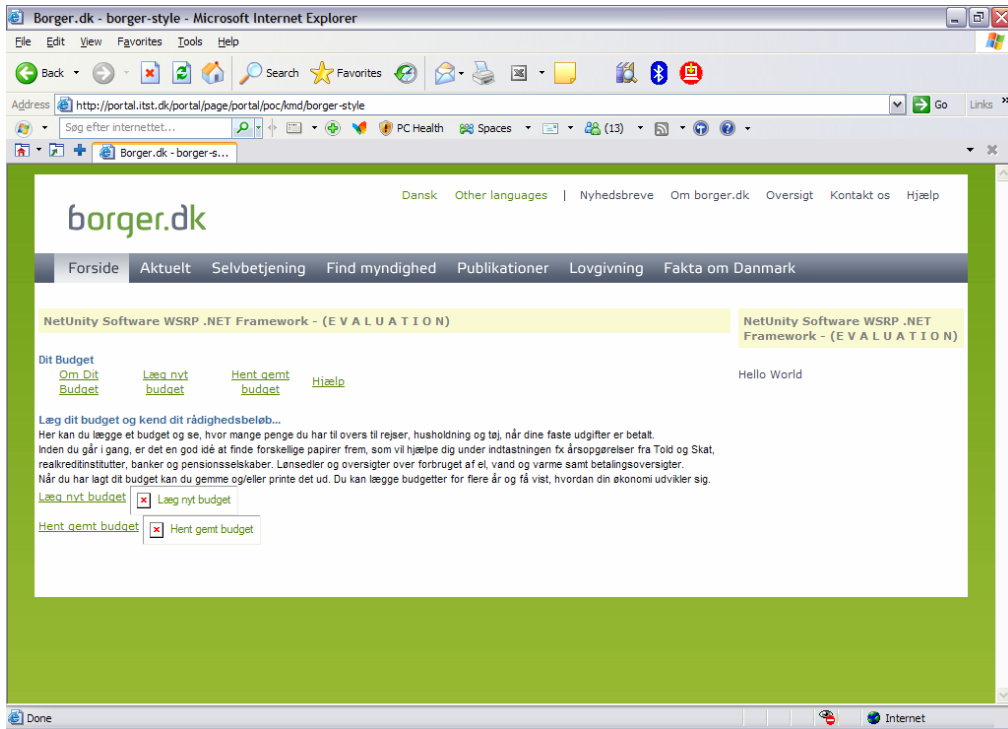
Remote Portlets er som andre portlets afgrænsede og konfigurerbare web komponenter, som i sin natur er tænkt til visuel integration. Portlets vises som én del af en samlet portal-side. Det er portalejeren, der bestemmer placering og størrelse, men serviceproviders design er afgørende for om resultatet er brugbart.

Selvom portlets afvikles hos serviceprovider er det muligt at lade portalejeren bestemme layoutet via styling. WSRP portlets har en række standard class names<sup>2</sup>, som portalen kan forvente at kunne bestemme style for. Se nedenfor eksempel på, hvordan samme portlet vises med to forskellige styles og dermed tilpasses hhv. et borgerportalslayout og et erhvervsportalslayout – uden at serviceprovider har rettet noget som helst:

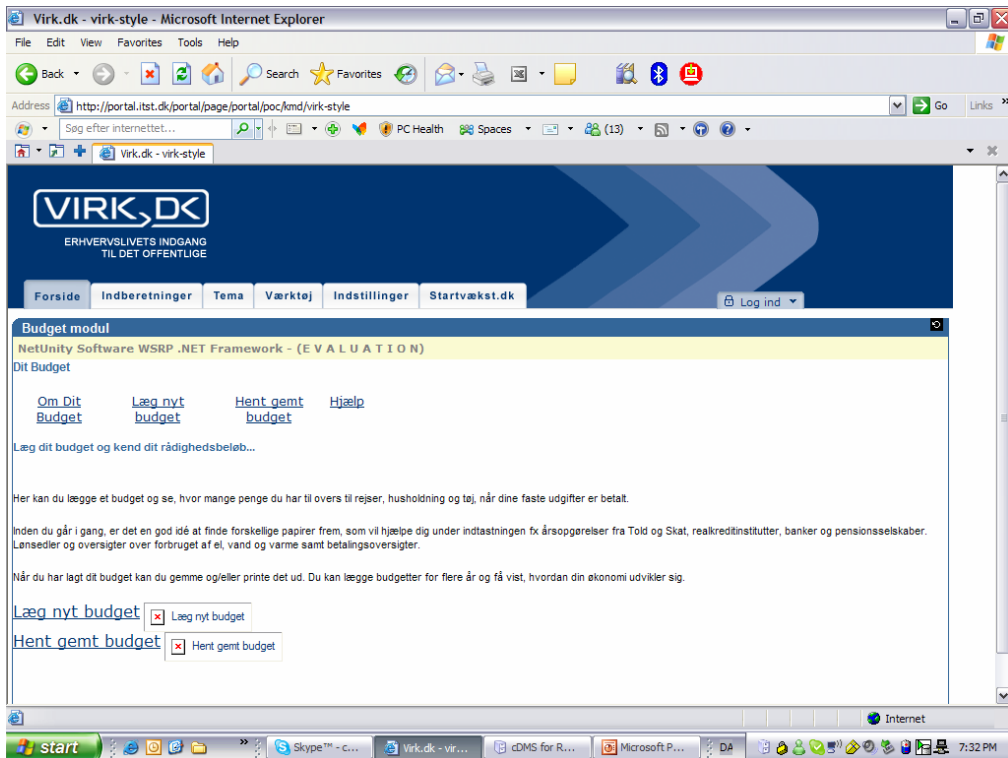
---

<sup>2</sup> I HTML kan angives en attribut *class* som er bestemmende for hvilken style et HTML element vises med (Eks.: `<div class="portlet-font" ...>abc</div>`)


Figur 2: Mit Budget med borger-style




Figur 3: Mit budget med erhvervs-style



Såfremt serviceprovider og consumer begge overholder standard navngivningen er det relativt banalt at lade en portlet styre efter portalens ønske.


	Det anbefales at Remote Portlets, der tænkes placeret på Borger- og Virksomhedsportalen <b>skal</b> overholde brug af standard class names og kun selv styrer style, hvor der er et særligt behov for at afvige standard styling for portalen.
---	--

Det er uproblematisk at tilpasse Portlets i størrelse og dermed lade portalejeren styre sammensætningen af skærbilledet. Vær dog opmærksom på, at serviceproviderens valg af design om brug af eksempelvis store grafiske elementer er afgørende for, om resultater bliver som man kunne ønske sig, hvis man ændre på den plads, der afsættes til en portlet.


	Det anbefales at der etableres nogle generelle designretningslinier, der gør, at remote portlets faktisk optræder ensartet – eksempelvis hvad angår knapper, under-menuer, progress bars mv.
---	--

#### 4.3.4 Mulighed for interaktion med andre services

Remote Portlets kan sammensættes med – og sameksistere med - andre portlets uden problemer. Det er afprøvet og de enkelte portlets holder fortsat deres tilstand, når man foretager handlinger i de øvrige. Det er dog afgørende at man overholder specifikationen retningslinier for navngivning, så der ikke opstår overlap mellem navne, der skal være unikke.

	Husk - af hensyn til sameksistens med andre portlets – at sikre unikke ID'er og funktionsnavne (i scripts).
---	---

Problemstillingen skyldes at portalen sammensætter dokumentet inden det sendes til browseren. Det er en anden teknik end ved iFrames, der opfattes som selvstændige HTML dokumenter som brugerens browser henter og selv sætter sammen.

	Vær opmærksom på, at da Portalen kan forventes at anvende HTML tables til at sammensætte skærbilledet af en række portlets, kan dette være i strid med tilgængelighedskrav (krav om at tables anvendes hvor indholdet logisk kræver brug af tabeller).
---	--


Der er ikke i WSRP 1.0 understøttelse af kommunikation *mellem portlets*. Dvs. at man kan – som nævnt ovenfor – uproblematisk sammensætte et skærbillede af flere portlets fra forskellige myndigheder, men man kan ikke lade handlinger i én af disse påvirke, hvad der sker i en af de andre. Det forventes dog at bliver understøttet i version 2.0. Det svarer altså til begrænsningerne i iFrames, men giver ikke de samme muligheder som JSR168 portlets.

Remote portlets giver ingen begrænsninger i forhold til kald af underliggende services, eftersom portleten stadig kører hos serviceprovider. Alle kald til back end og træk på legacy er altså uændret. Dette er en væsentlig forskel i forhold til traditionelle (JSR168) portlets, som installeres på portalen.

#### 4.3.5 Performance og driftsmæssig tilgængelighed

Brugen af remote portlets giver et rent snit i forhold til ansvaret for udvikling og drift, eftersom portlets fortsat driftsmæssigt befinder sig hos serviceprovider. Det er en klar forsimpning i forhold til at en myndighed skal have kode installeret på portalen, hvor der kan opstå kompatibilitetsproblemer og forretningsmæssigt være problemer med at placere et ansvar for evt. problemer. Det betyder dog at afhængigheden af netværket og båndbredde er stigende (i forhold til installerede portlets).

WSRP portlets kaldes over en Web Service. Det giver et overhead dels pga. at kaldet sker over netværk til en anden myndighed og dels fordi web services er baseret på XML og SOAP. Det er dog konstateret at portalsider med WSRP portlets er i stand til at svare pænt under 1 sekund. Samtidigt er der uden egentligt load test konstateret at flere remote portlets på samme side med flere samtidige kald øger denne svartid. Der er ikke gennemført egentlig load test i PoC og hardware er ikke skaleret efter at skulle performe, men det er stadig en indikation af, at performance skal have opmærksomhed ved valg af denne integrationsform.

	<p>Vær opmærksom på, at der er behov for et miljø, hvor man kan give udefrakommende adgang til at kalde web services samtidigt med, at man selv ofte skal kunne nå nogle back end (legacy) applikationer.</p> <p>Allerede under test bliver dette relevant og interne testmiljøer er sjældent tilgængelige udefra. Det bør tidligt i processen tænkes, at dette behov til opstå, da drift- og sikkerhedshensyn kan gøre det vanskeligt at få de rette ressourcer stillet til rådighed.</p>
---	--

#### 4.3.6 Øvrige erfaringer

WSRP specifikationen indeholder nogle user attributes (name, e-mail, phone mv.) som kan bruges til at give information om *slutbrugeren*. Det er ikke den bruger serviceprovider taler sammen med i sikkerhedsmæssig forstand (det er typisk en systembruger). Disse attributter kan dog i sammenhæng med en sikker forbindelse mellem serverne anvendes til at overføre eksempelvis et CPR nummer og dermed kombineret med den sikre linie og en aftale om hvordan portalen skal sikre autenticitet af brugeren i forhold til CPR anvendes til at hente personaliserede og evt. følsomme oplysninger fra serviceprovider.

Det må ikke forveksles med den sikkerhedsløsning hvor SAML tokens giver sikker identifikation af slutbrugeren. I ovennævnte scenarie taler system-system sammen og udveksler blot nogle attributter. Serviceprovider har intet bevis for at slutbrugeren faktisk er autentificeret og af hvem.

#### 4.3.7 Opsummering: Fordele og ulemper ved denne integrationsform

Fordele +	Ulemper -
<ul style="list-style-type: none"><li>• Giver en tæt visual integration med mulighed for at lave portalsider bestående af services fra forskellige myndigheder på en måde så de fremstår sammenhængende</li><li>• Remote Portlets giver et klart snit mellem portal og den ansvarlige myndighed som selv har såvel udvikling som drift af servicen, der blot vises på portalen</li><li>• Platformsuafhængighed. Da WSRP er baseret på Web Services er der platformsuafhængighed og i praksis har det vist sig relativt let at arbejde på tværs af produkter fra forskellige leverandører</li><li>• Det er sikkerhedsmæssigt at foretrække at brugerens browser kommunikerer med én portal, frem for som ved iFrame med en række web servere</li><li>• Services, der har behov for kald af back end systemer, giver ingen problemer, da de ikke skal installeres på portalen og dermed stille krav om at der fra portalen åbnes op for diverse (nye) integrationer</li></ul>	<ul style="list-style-type: none"><li>• En SSO løsning med samspil mellem WSRP og SAML er ikke klar. Sikkerhedsløsningen skal derfor baseres på en taktisk løsning, indtil en profil for SAML for web services er på plads.</li><li>• WSRP 1.0 understøtter ikke inter-portlet kommunikation. Det forventes dog at komme i version 2.0</li><li>• Standarden er relativ ny og fortsat under udvikling.</li><li>• Der kan være relativt omfattende omskrivning af legacy applikationers front end pga. form elementers navngivning, brug af javascript samt brug og eks. cookies</li></ul>

## 4.4 Composite Application - JSR168 portlets

Portlets baseret på JSR168 er afgrænsede web komponenter som afvikles i portalens run time miljø. Portlets kan anvendes både til simple og til sammensatte applikationer.

### 4.4.1 Integrationsformen og standarder

Portlets er baseret på JSR168 standarden. Der er tale om en moden standard fra 2003, der beskriver hvordan man i java koder en portlet. JSR168 er ikke en del af J2EE. Der er en ny version af JSR på vej, men den er endnu ikke klar og produktunderstøttet.

Microsoft har deres eget bud på en konfigurerbar web komponent; web part og understøtter ikke JSR168.

Før JSR168 havde en række java leverandører hver deres bud på portlets. For interoperabiliteten er det afgørende, om man holder sig indenfor standarden.

### 4.4.2 Produktunderstøttelse og interoperabilitet

JSR168 er understøttet af en række leverandører i Java verdenen. Bemærk dog at JSR168 er en *java standard* og som sådan ikke understøttet af Microsoft (og flere andre). Microsofts analoge teknologi er web parts. I PoC konstateredes det, at Oracle Portal har implementeret JSR168 understøttelse ved at indpakke og afvikle disse som WSRP.

JSR168 kan udvikles på én platform og herefter afvikles på én anden (java) platform. Interoperabilitet er meget lige til så længe portlets er meget simple. En hello world portlet tager ganske kort tid at flytte over på en anden java platform. Når man kommer nærmere virkelighedsnære og mere komplekse løsninger var det dog knap så trivielt at installere på ny platform. Det skyldes ikke standarden og understøttelsen af denne i sig selv, men at portlets *installeres* på portalen og dermed bliver afhængig af portalens run time miljøer og den infrastruktur som hele platformen tilbyder.

Det kan give problemer, selv hvor begge platforme *har* implementeret understøttelse af en standard, der anvendes, eller begge faktisk tilbyder en bestemt infrastrukturkomponent som et web service runtime miljø, fordi der er tale om forskellige implementeringer og en JSR168 portlet kan være udviklet på en måde, der binder den tæt til den enkelte implementering.



Vær opmærksom på, at JSR portlets, som installeres på portalen kan være afhængige af dele af den infrastruktur, hvor den er udviklet!


Ligesom mange andre Java applikationer kan en JSR168 portlet anvendes diverse java biblioteker, til at løse specielle opgaver i applikationen. Hvis disse biblioteker ikke er understøttet i portal platformen, skal de pakkes ind i deployment filen, som indeholder JSR168

portleten. Derved bliver de også tilgængelige, på consumer portalen. Det er almindelig "god praksis", men udfordringen er at finde den rette balance så hver portal ikke medtager unødvendigt mange biblioteker og ignorerer at portalen allerede understøtter meget af den infrastruktur, den måtte have behov for. Der vil blive behov for retningslinier og aftaler.

#### 4.4.3 Mulighed for styring af brugergrænsefladen

Portlets er afgrænsede og konfigurerbare web komponenter som i sin natur er tænkt til visuel integration. Portlets vises som én del af en samlet portalside. Det er portalejeren, der bestemmer placering og størrelse, men serviceproviders design er afgørende for om resultatet er brugbart.

Da portlets afvikles på portalen er det muligt at lade portalejeren bestemme layoutet via styling. JSR168 portlets har en række standard class names<sup>3</sup>, som portalen kan forvente at kunne bestemme style for. Se eksempel under Remote Portlets, der har samme mulighed.

	Det anbefales at Portlets, der tænkes placeret på Borger- og Virksomhedsportalen <b>skal</b> overholde brug af standard class names og kun selv styrer style, hvor der er et særligt behov for at afvige standard styling for portalen.
---	---

Såfremt serviceprovider og consumer begge overholder standard navngivningen er det relativt banalt at lade en portlet style'e efter portalens ønske.

Det er uproblematisk at tilpasse Portlets i størrelse og dermed lade portalejeren styre sammensætningen af skærbilledet. Vær dog opmærksom på, at serviceproviderens valg af design om brug af eksempelvis store grafiske elementer er afgørende for, om resultater bliver som man kunne ønske sig, hvis man ændre på den plads, der afsættes til en portlet.

#### 4.4.4 Mulighed for interaktion med andre services

Portlets giver i modsætning til 1. generation af Remote Portlets mulighed for inter-portal kommunikation. Portlet-interaktion er dog ikke afprøvet i PoC, men portlets er velafprøvede i en række andre sammenhænge.

Portlets kan uproblematisk sameksisterer med andre portlets (og hver for sig "holde tilstand"), når blot der overholdes nogle retningslinier for navngivning. Portlets og Remote Portlets kan tillige sameksisterer. Der er ikke konstateret nogen problemer i denne sammenhæng under PoC.

Såfremt Portlets anvender back end systemer eller har behov for at hente data fra en database eller lignende opstår der et noget mere kompliceret billede. Idet portlets er installeret på portalen, betyder det, at enten skal der også installeres databaser mv. på portalen

---

<sup>3</sup> I HTML kan angives en attribut *class* som er bestemmende for hvilken style et HTML element vises med (Eks.: `<div class="portlet-font" ...>abc</div>`)

eller disse portlets skal have lov til (eks. via JDBC) kunne kalde en database over netværk. På samme måde kan der være behov for at kalde andre services via web services eller andre metoder. Det betyder, at der i realiteten åbnes op for helt andre integrationsformer, som giver et langt mere kompliceret billede af, hvad der ligger på portal og hos myndighed og hvordan samspillet er herimellem.



Vær opmærksom på, at JSR portlets installeres på portalen og derfor skal der fra portalen være adgang til eventuelle back end systemer.

Denne adgang kan være baseret på integrationsformer, der ikke tillades eller er forbundet med tekniske og sikkerhedsmæssige problemer.

Såfremt man måtte fravige princippet om en tynd portal og samtidigt etablerer en offentlig Serviceorienteret Infrastruktur med Web Service kataloger, offentlige service-busser mv. tegner der sig et mere positivt billede af at anvende JSR168 portlets, som installeres på portalen og udnytter en sådan offentlig service infrastruktur.

#### 4.4.5 Performance og driftsmæssig tilgængelighed

Grundlæggende er der tale om et mindre "rent snit" mellem serviceprovider og portal, idet JSR portlets installeres på portalen. Der er altså tale om at man udvikler kode ét sted og installerer et andet. Der er tillige tale om, at driften ligger på portalen, mens myndigheden bag fortsat må have det forretningsmæssige ansvar.

Rent performancemæssigt vil JSR portlets forventes at være bedre end Remote Portlets, eftersom der er tale om installeret kode og ikke et web service kald over netværk. Det er dog afgørende at der ikke blot flyttes netværkskald til et andet lag (hvis den installerede portlet herefter kalder services "hjemme hos myndigheden" over netværk). Desuden spares SOAP/XML overhead'et også.



Man kan overveje om JSR168 portlets er oplagte til *fælles/interne* services på portalen, mens de er mindre oplagte til integrationer mellem myndigheder – pga. fordelene i performance kontra ulemperne i forhold til drift og ansvar.

#### 4.4.6 Øvrige erfaringer

Oracle Portal understøtter JSR168 via en WSRP indpakning. Det betyder at JSR168 portlets må forventes afviklet så WSRP begrænsningerne/ulemperne også er gældende for JSR på netop denne platform!

#### 4.4.7 Opsummering: Fordele og ulemper ved denne integrationsform

Fordele +	Ulemper -
<ul style="list-style-type: none"><li>• Giver en tæt visual integration med mulighed for at lave portalsider bestående af services fra forskellige myndigheder på en måde så de fremstår sammenhængende</li><li>• Der er rigere mulighed for at opbygge sammenhængende applikationer og der er mulighed for inter-portlet integration (i modsætning til WSRP)</li><li>• Performance er bedre end for Remote Portlets (under forudsætning af at man ikke flytter en række netværkskald fordi portleten er afhængig af andre services)</li></ul>	<ul style="list-style-type: none"><li>• Ikke platformsuafhængighed. JSR portlets er kun en java standard. Det er ikke en platformsuafhængig standard som også Microsoft og andre understøtter</li><li>• Portlets skal installeres på portalen og kan give udfordringer med at arbejde på tværs af platforme, fordi de er udviklet et sted og afhænger af den infrastruktur der var tilstede under udvikling</li><li>• Kald af back end og legacy er problematisk, fordi der skal åbnes op fra portalen</li><li>• En SSO løsning baseret på SAML omfatter ikke kald af back end systemer (hvor disse måtte have behov for at kende slutbrugeren)</li></ul>

## 5 Konklusion

Der er afprøvet en række integrationsformer baseret på forskellige standarder. Alle services er integreret og har virket under afprøvning. Så grundlæggende kan alle integrationsformer anvendes fra 2008 med de nuværende standarder og produktunderstøttelser.

Der er dog forskellige udfordringer, der kan forventes i forbindelse med udvikling og praktisk anvendelse af disse. Se nærmere om dette under afsnit 4 omkring afprøvning og erfaringer.

Generelt er *ingen* af de afprøvede integration uproblematisk. Links er klart den mest simple integrationsform, men brugen af SAML er ganske nyt for de fleste leverandører og ikke mindst danske implementeringspartnere og for nuværende findes den fælles logintjeneste ikke. Der er alene opsat en IdP i PoC sammenhæng.

JSR168 er en relativ velafprøvet standard, men så snart man bevæger sig på tværs af platforme (Microsoft .NET helt ude) og myndighedsinstallationer kan billedet kompliceres – lige som den integrationsform er den *mindst* ønskværdige i forhold til "glaspladeprincippet" og en klar opdeling mellem myndigheds- og portalejers ansvar.

WSRP er en relativ uafprøvet standard og der forventes indenfor overskuelig tid en ny version. Der er ved brug af WSRP behov for taktiske sikkerhedsløsninger (som dog ligner hvad der findes i dag i kørende portalløsninger).

I det følgende opsummeres nogle væsentlige budskaber i forhold til om en integrationsform kan betragtes som anvendelig.

### Genbrug af eksisterende løsninger

Links og iFrame er nærliggende til at videreføre eksisterende web løsninger. Bemærk dog at brugen af SSO teknologi ikke er helt trivial.

Har man en java platform vil JSR168 og WSRP portlets oftest også være en oplagt mulighed.

Har man udviklet på Microsoft kan JSR udelukkes mens såvel link, frame som WSRP fortsat er kandidater. WSRP vil dog her være lidt mindre trivial.

### Tæt visuel integration?

Hvis der ønskes en tæt visuel integration mellem myndighed og portal og evt. med services fra forskellige myndigheder bør der anvendes Frames eller Remote Portlets. JSR168 portlets kan anvendes, men det vil bryde med glasplade princippet.

Er der ønske om at afsender fortsat har egen identitet er Link m/SSO oplagt.

### **Interaktion mellem myndighedsservices**

Er det behov for at forskellige services på portalen taler sammen, er det afgørende om man kan vente på WSRP 2.0 som forventes at understøtte dette.

Såfremt man har behov for interaktion *mellem* services fra forskellige myndigheder her og nu er der kun JSR168 portlets (eller work arounds udenfor standarderne som ikke er afprøvet i PoC sammenhæng).

Myndigheder kan dog – uafhængigt af portal integration – altid opbygge serviceorienterede løsninger, hvor myndighederne selv implementerer en integration og *herefter* integrerer slutresultatet op mod en af portalerne. Det er hvad case'en med Færdselsstyrelsen viser.

### **Sikkerhed**

Såfremt sikkerhed er et væsentligt emne for en integration, skal man afklare om en taktisk løsning er tilstrækkelig. Det kan være en taktisk løsning med eller uden brug af SAML *tokens*, men kræver protokol og profil, der går udover hvad SAML og danske profiler byder på i dag. Ellers skal man basere løsningen på Link eller Frame med SSO indtil videre. Det forventes at ændre sig indenfor kort tid hvor man må vente at en SAML profilering til andre services er klar.