

Governance kræver en beholder til meta- data

Whitepaper af Henrik Hvid Jensen

12. september 2006

Indholdsfortegnelse

Governance kræver en beholder til metadata	2
Et UDDI-register i forhold til en Metadata-beholder	4
SOA-styrings-politikker indfanger metadata	5
Overholdelse af kontrakter/politikker på kørselstidspunkt.....	6
Konsekvensanalyse	7
Troværdighed og tillid.....	9
Håndtering af livscyklus.....	9
Opsummering.....	10

Governance kræver en beholder til metadata

De tidlige visioner om SOA og webservices lovede, at leverandører af services kunne offentliggøre, hvorledes forbrugeren kunne tilgå funktionaliteten, i et UDDI-register. Potentielle forbrugere kunne finde servicen og starte med at bruge den uden overhovedet at interagere med leverandøren. I denne løst koblede model behøvede applikationsudviklere ikke nødvendigvis at kommunikere med hinanden for at forbinde deres applikationer. Ved brug af åbne standarder kan forbindelse praktisk talt være anonym.

Men efterhånden som virksomheder er begyndt at benytte teknologien som alternativ til traditionelle applikations-integrationer og B2B-integrationer, har de oplevet et behov for ikke kun at styre teknologien, men også styre forretningsrelationerne med partnere og forretningsenheder. Der er derfor behov for administrations- og kontrolprocesser til at:

- autorisere idriftsættelse og adgang
- styre relationerne mellem leverandør og forbruger
- overvåge brugen af services
- forstå hvilke applikationer der forbruger hvilke services
- sikre tilfredsstillende tilgængelighed, troværdighed, forudsigelighed og performance
- sikre konsistens og servicekvalitet
- holde styr på forbrug og afregning
- håndhæve brug af forretningstermer fra virksomhedens begrebsmodel
- osv.

Hvis services skal blive seriøse komponenter i en virksomheds it-strategi, er det nødvendigt at forstå, at SOA ikke bare er en samling af webservices. SOA har en meget *bredere* synsvinkel. SOA er en struktureret og planlagt tilgang til design, udvikling og integration af forretningsorienterede services på tværs af forretningsområder og organisatoriske grænser.

Governance (it-styring) er de regler, processer og praksis, der beskriver, hvem der tager hvilke beslutninger hvornår. Det gøres gennem politikker udstukket af ledelsen. Disse politikker beskriver et antal aktiviteter, som en organisation skal følge for at implementere en effektiv it-styring. Virksomheden implementerer politikkerne som forretningsprocesser med beslutningspunkter såsom inspektion/review af designet og de roller, som personer skal tildeles for at være ansvarlig for at tage specifikke beslutninger.

SOA introducere nye udfordringer omkring it-styring. SOA indeholder mange uafhængige og selvstændige komponenter, som er genbrugt på tværs af virksomheder og er en vital del af forretningskritiske forretningsprocesser. Styringen af virksomhedens SOA skal sikre, at man ved:

- Hvad der sker når en service ændres?
- Hvordan man kan være sikker på, at den service man bruger har en høj kvalitet?
- Hvad der sker, hvis en delkomponent af en sammensat service tages ud?
- Hvordan man kan være sikker på at en ny service overholder it-, forretnings- og lov-krav?
- Hvordan man kan sikre en forudsigelig opetid på en service?
- osv.

Disse behov er ikke nye, men det drejer sig om at mikse den serviceorienterede fleksibilitet med den traditionelle kontrol af it-arkitekturer.

Udfordringen til at styre virksomhedens SOA ligger i at levere en tilstrækkelig governance-infrastruktur uden at sætte behændigheden og fleksibiliteten af arkitekturen på spil. Hvis virksomheden vælger at fasttømre deres governance-værktøjer og processer, vil den miste løftet om behændighed. Det er derfor nødvendigt at bygge fleksibilitet ind i selve governance-infrastrukturen.

Hemmeligheden til at vedligeholde behændigheden ligger i metadata. Metadata er information om data og i denne situation information omkring it-artefakter såsom forretningsprocesser, forretningsregler, servicebeskrivelser og ethvert andet element i it-miljøet. Metadataene udgør f.eks. navnet på artefaktet, placeringen, vigtigheden, kvaliteten eller værdien for virksomheden og dets relationer med andre artefakter som en virksomhed vurderer, er værd at styre (se også Figur 1 på side 5). Metadata om it-aktiverne tilbyder information, der muliggør effektiv styring af software-aktiver gennem hele deres livscyklus

SOA kræver, at metadata er placeret eksternt fra applikationen, der implementere services for at muliggøre klassifikation og styring af disse uafhængige services. Evnen til at håndtere metadata udgør livsnerven i enhver SOA-implementering. Man bruger metadata til at tage beslutninger med. Hvis beslutningstageren har relevant metadata til rådighed, tager han bedre og mere præcise beslutninger

Mange virksomheder forsøger at håndtere denne metadata ved at bruge værktøjer såsom regneark, word-dokumenter eller Visio-diagrammer. For effektivt at kunne styre og håndtere mere komplekse systemer er det nødvendigt med en discipline-

ret proces til styring af metadata, der er understøttet af et Metadata Repository (MR) (Metadata-beholder).

Et UDDI-register i forhold til en Metadata-beholder

Et UDDI-register er en nøgle-komponent i en SOA-infrastruktur. Men efterhånden som virksomheder bevæger sig mod større implementeringer, er der behov for:

- En SOA-informationsmodel baseret på virksomhedens begrebsmodel
- En Metadata-beholder der opbevarer metadata til brug ved governance
- Governance regelmaskine
- Rig servicebeskrivelse og dokumentation

Man vil sjældent bruge metadata-beholderen på kørselstidspunktet. Det skyldes dels større sikkerhedsudfordringer, dels at de typisk har et større antal artefakter opbevaret end et UDDI-register, der bruges på kørselstidspunktet. De er derfor typisk ikke optimeret til at fungere effektivt i et driftsmiljø og er designet til et andet publikum end et register.

Metadata-beholderen kan man betragte som et slags metadata-varehus, hvorfra man styrer, forstår og rapporterer om væsentlige metadata. Det eliminerer ikke behovet for at vedligeholde metadata, som understøtter specifikke udviklings- og kørsels-aktiviteter. Et register fokuserer på at styre specifikke delmængder af metadata i kørsels-miljøer i flere detaljer end beholderen.

En beholder vil ikke gemme alle metadata, som relaterer sig til f.eks. BPM, workflow eller database design og idriftsættelse. Det gemmer i stedet en delmængde af information for hver. Figur 1 illustrerer typisk indhold i et register i forhold til en beholder.

	Data	Funktion	Timing	Område	Motivation
Kontekstuel	Oversigt over ting der er vigtige for forretningen	Oversigt over processer som virksomheden deltager i	Oversigt over hændelsesforløb der er vigtige for virksomheden	Oversigt over områder som virksomheden operere i	Oversigt over vigtige forretningsmål og strategier
Konceptuel	Semantisk model	Model for Forretningsproces	Overordnet skemalægning	Logistik system	Forretningsplan
Logisk	Logisk data-model	Systemarkitektur	Aktivitetsmodel	Distribueret systemarkitektur	Model for forretningsregler
Fysisk	Fysisk data-model	Applikationsarkitektur	Kontrolstruktur	Teknisk arkitektur	Design af forretningsregler
Implementering	Implementeret datamodel	Kode arkitektur	Implementering af kontrol	Idriftsat arkitektur	Implementering af forretningsregler

Repository / Beholder (left arrow) and Register (right arrow) are indicated by large light blue arrows pointing towards the table.

Figur 1 Eksempler på indhold i et register i forhold til en beholder (Kilde Gartner (November 2005))

Som figuren illustrerer, er der ikke en klar adskillelse mellem de metadata, der opbevares i hver. Overordnet tager registeret udgangspunkt i behovet på kørselstidspunktet, mens beholderen tager udgangspunkt i behovet på designtidspunktet. Beholderen indeholder metadata til rapportering og konsekvens-analyser, mens et register fokuserer på at håndtere metadata relevant på kørselstidspunktet. Et register svarer til kortindekset over bøgerne på et bibliotek. Indekset peger på den bog, som en læser ønsker, på samme måde som et UDDI-register peger på den service, en service-forbruger ønsker at tilgå. Beholderen håndterer metadata om en bredere gruppe af artefakter. Den håndterer metadata om biblioteker, kortindekser, bøger og processer for at låne en bog og aflevere den igen.

Figuren viser også at forskellige roller i organisationen skal bruge forskellige delmængder af metadata på tværs af miljøer og behov.

SOA-styrings-politikker indfanger metadata

De SOA styrings-politikker virksomheden etablerer, definerer konfigurerbare regler og betingelser, som har indflydelse på både design-tidspunkt og kørselstidspunkt. Det betyder, at virksomheden skal bruge politikker til at validere services, før de er offentliggjort (på designtidspunktet) og som et fundament til at sikre specifikke standarder og opførsel på kørselstidspunkt.

Politikkerne indfanger metadata såsom de dokumenter, der bliver produceret i forbindelse med inspektionen af kravspecifikationen, design og kodning samt informationen om hvornår det skete og hvem der godkendte inspektionen.

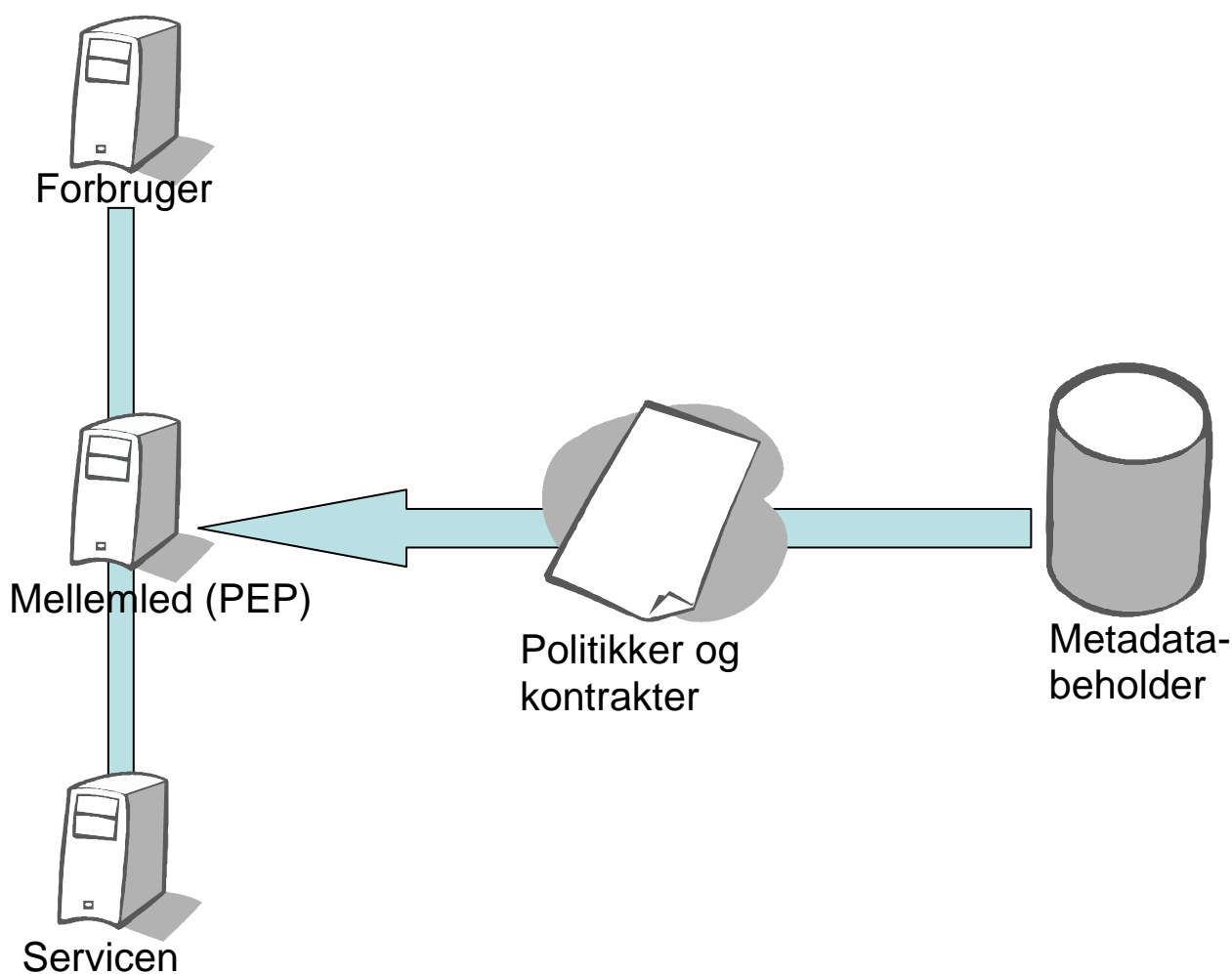
Tæt koblede systemer definerer governance og kontrol i forbindelse med applikationen. SOA er anderledes, fordi applikations-konteksten er variabel og evigt skiftende. Det betyder, at governance må håndteres på et andet abstraktionsniveau, på selve servicen. Politikkerne skal derfor tages ud af koden og placeres eksternt som metadata til servicen.

Servicereglerne skal være dannet baseret på standardiserede, genbrugelige politikker, som kan blive tilknyttet services. En sådan forbindelse mellem en service og politik muliggør automatisk validering af servicen og overholdelse af specifikke politikker. En opgave for virksomheden er derfor at gøre de nuværende papirbaserede politikker til elektroniske regler, så processen for validering og overholdelse kan automatiseres.

Overholdelse af kontrakter/politikker på kørselstidspunkt

Politikker bliver ofte tildelt ensartet på tværs af en større gruppe af brugere, hvormed det er muligt at håndtere flere serviceforbruger ens. Kontrakter er typisk håndteret af politikker og kan betegnes som instanser af politikker – services er leveret i henhold til forbrugers præferencer, men indenfor de begrænsninger, som organisationens politikker sætter. I mange forretningsforhold repræsenterer kontrakter aftaler, derfor vil kontrakter ændre sig efterhånden som leverandør/forbruger relationerne ændres.

Overholdelse af politikker for flere services på kørselstidspunkt kan ske via et mellemlid, der fungerer som et Policy Enforcement Point (PEP) (Figur 2). Muligheden for et sådant fælles mellemlid er det direkte resultat af standarderne for beskudveksling. Metadata-beholderen er central for at opbygge og opbevare konfigurerbare politikker. SOA-politikker bør være løst koblet og centralt styret



Figur 2 Politikker opbevares i Metadata-beholderen og overholdes af et mellemlid PEP

Konsekvensanalyse

Virksomheders behov for hurtige ændringer er en af drivkræfterne bag SOA. Nøglen til forretningsbehændigheden i en SOA er muligheden for forretningsbrugere til fleksibelt at tilpasse SOA-løsningen til ændrede markedsbehov. I en løst koblet verden vil der derfor konstant ske ændringer. Arkitekturen kan bestå af mange hundrede services, som udvikler sig og ændrer sig baseret på deres egen unikke livscyklus. Den behændighed og tættere tilknytning til forretningen, som SOA giver, betyder at en del af SOA-governance er at identificere og håndtere konsekvenserne og afhængighederne af en ændring.

En bivirkning ved at opdele enkeltstående systemer i mindre håndterbare services, er en forøgelse i det samlede antal applikationer, der skal styres. Virksomheder skal derfor kunne analysere, forstå og planlægge ændringer på tværs af forretningsprocesser, organisationer, services og underliggende infrastruktur.

Det er nødvendigt, at virksomheden strukturerer metadata om virksomheden og dens it-aktiver, for at den kan opnå denne forståelse.

Der er specielt et antal ændringsområder, som virksomheder står overfor i dag.

- **Servicekontraktens metadata** - Den løse kobling giver udviklerne mulighed for at ændre implementeringen af en service uden af forbrugeren af servicen skal ændres samtidig (og vice.versa). Men det betyder ikke, at ændringer til en service ikke har indflydelse på andre parter. Den løse kobling holder kun, hvis den underliggende implementering stadig overholder og respekterer den eksisterende servicekontrakt. Så selv små ændringer til selve servicekontrakten kan derfor give problemer. Virksomheden skal sikre, at ældre versioner af en servicekontrakt forbliver gyldige, indtil organisationen, *gennem god governance*, kan sikre, at ingen forbruger vil anvende disse kontrakt-metadata igen.
- **Versionering af politikker** – Som tidligere omtalt er det vigtigt at politikker kan vedligeholdes separat fra selve service-kontrakten. En sådan uafhængig ændring vil forøge kompleksiteten af SOA-implementeringen. På et vilkårligt tidspunkt kan en person ændre en politik, der har indflydelse på et ukendt antal services. For at undgå uønskede konsekvenser ved at ændre politikker, bør virksomheden implementere metapolitikker, som kontrollerer, hvem der kan ændre politikker og hvornår de kan gøre det. Der skal etableres tilstrækkelig governance for at kontrollere, hvordan virksomheden danner, kommunikerer og sikrer overholdelse af politikker. Det er kun autoriserede personer, der kan ændre politikker og det må de kun gøre, når der er tilstrækkelig forståelse for konsekvenserne. For at opnå succes i SOA-implementeringen skal virksomheden *tidligt i SOA-projektet* etablere et governance-rammeverk.
- **Begrebsmodellen** – Selv om hverken kontrakten eller politikkerne ændres, kan de underliggende data-skemaer godt ændres. Det er kun muligt at opnå et acceptabelt niveau af løs kobling, hvis både service-leverandøren og service-forbrugeren er enige om en fælles forståelse af de data, de kommunikerer. Services bør afhænge af vokabularer og/eller ontologier, som alle parter involveret i en service-orienteret dialog deler. Styringen af skemaerne skal ske samtidig med styring af andre service-metadata. Data-skemaer og semantik bør være en kritisk del af den metadata, der er nødvendige for service-interaktioner. Hver gang et skema ændres, skal alle relevante services checkes for at sikre, at de stadig overholder deres servicekontrakter. *Governance skal derfor inkludere styring af data-skemaer og begrebsmodeller.*

Disse udfordringer er ikke nye, de har typisk været håndteret i virksomhedens enkelte integrationsprojekter og derved været tæt koblet til isolerede specifikke systemer. SOA-governance giver mulighed for en *fælles tilgang* for at håndtere ændringer og dette ikke kun på basis af hver system-til-system løsning men som en

systematisk tilgang, som giver virksomheden mulighed for at håndtere metadata-ændringer *uafhængigt* af det underliggende system, proces eller applikation.

Virksomheder må forstå, at ændringer er uundgåelige og må derfor planlægge hvordan disse ændringer håndteres.

Troværdighed og tillid

Det kræver tillid at opbygge et effektivt miljø til deling af services. Serviceforbrugerne vil ikke genbruge services, hvis de ikke kan blive forsikret om kvalitet, forudsigelighed og fælles forståelse af begreber og betingelser. På samme måde bør organisationer ikke opfordre til brug af services uden at forstå og kontrollere adgang, provisioning (tildeling af adgang til data og teknologiske ressourcer) og forstå den samlede tilstand af de genbrugelige services.

Servicekontrakter er vigtige arkitekturværktøjer for at kommunikere og overholde politikker og andre krav i et heterogent og distribueret it-miljø. Ligesom forretningskontrakter sikrer et sundt økonomisk samarbejde vil en servicekontrakt sikre et sundt leverandør/forbruger samarbejde og hjælpe med at etablere en aftale og opretholde tilliden mellem partnerne.

Det kræver, at aftalen tilbyder en præcis og entydig aftale for hvordan leverandøren og forbrugeren interagerer. Kontrakter er typisk unikke for en specifik leverandør/forbruger relation og fungerer som opbevaringssted for både formelle politikker såvel som aftaler, der er unikke for partnerne.

Da de er unikke for hver individuel leverandør/forbruger relation, er kontrakter typisk dannet på tidspunktet for forbrug af servicen. Men det betyder ikke, at de skal omskrives hver gang. Mange kontrakter kan og bør blive vedligeholdt og genbrugt for at danne en basis for mange aftaler. Derfor er kontrakter en anden vigtig SOA-artefakt, som virksomheden skal styre og genbruge.

Håndtering af livscyklus

Det er en iterativ proces at implementere services. Fokus på at levere servicen til forretningen afhænger derfor af processen, som understøtter udviklings- og leverance-cyklussen af den underliggende teknologi-forbedring. Politikker til at håndtere servicens livscyklus er derfor en kritisk del af SOA-governance og afhænger af effektiv håndtering af metadata om alle it-aktiver for at bibeholde fleksibilitet i henhold til ændrede forretningskrav.

En organisation, der bevæger sig frem mod en SOA, vil opleve, at antallet af aktiver, som en it-organisation skal styre, vil vokse. I traditionelle udviklingsprojekter hører kravspecifikationer, design-komponenter, kildekode, testplaner, procesmo-

deller, software- og data-arkitekturmodeller, softwaredesignmodeller, fysiske datamodeller, data-semantik, services, komponenter osv. til en specifik projektgruppe i en specifik afdeling. De enkelte afdelinger opbevarer typisk disse artefakter i forskellige værktøjer. I en SOA vil aktiver, der repræsenterer det samlede billede af servicens livscyklus, derfor typisk befinde sig *på tværs* af applikationer og afdelinger i organisationen. SOA-governance skal derfor understøtte, at virksomheden håndterer aktiver på tværs af organisationen, det gør den ved brug af metadata.

Udover metadata om det enkelte aktiv er metadata, der viser sammenhæng mellem aktiverne, væsentlig. For eksempel del-underdel relationer og hvordan services er relateret til arkitektur- og softwaredesignmodeller.

Metadata indeholder også information såsom datakvalitet, brug af data, hvor det er placeret, semantikken osv.

SOA livscyklus drejer sig om at:

- Sikre kvalitet, performance og relevansen af services, der er offentliggjort
- Tilbyde en måde for forbrugere at finde og genbruge services og andre artefakter
- Håndtere versioner af services
- Vurdere og håndtere konsekvenser ved ændringer på tværs af et netværk af forbrugere

Håndtering af en SOA-livscyklus inkluderer både individuelle services, efterhånden som de bliver designet, bygget og idriftsat, samt netværket af services, hvor servicen tilgås og bruges af forskellige grupper af serviceforbrugere

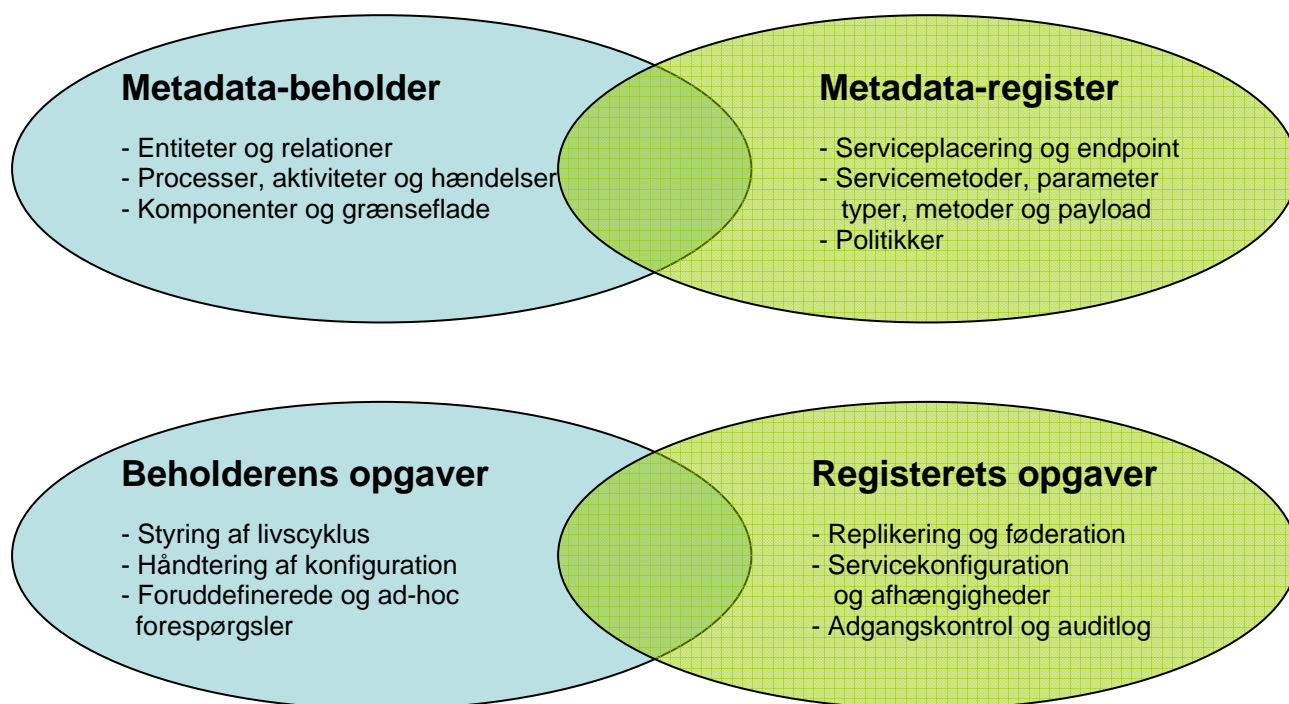
Opsummering

SOA forbinder services fra forskellige domæner, det er derfor nødvendigt med et system, som udviklere kan konsultere på design-tidspunktet. Det skal tilbyde mere information end et rent UDDI-register og imødegå et behov for at inkludere dokumentation og andet understøttende materiale. Der er behov for en metadata-beholder med dets egne definerede processer. Processerne skal ikke kun understøtte design-krav men også understøtte idriftsættelse, produktion og ændringer.

En metadata-beholder er værktøjet, når der er behov for at håndtere og forstå metadata, for at kunne styre artefakter, som går på tværs af såvel design- som kørsels-miljøer. En metadata-beholder er en central teknologi i en SOA. Det vil være vanskeligt at lave en succesfuld SOA på længere sigt uden en integreret og søgbar metadata-beholder.

Styring af metadata ud fra metadata-ejerens perspektiv er en del af governance-reglerne for virksomhedens arkitektur

Metadata-beholdere og registre tilfredsstillere hver deres type af brugere og formål. De fleste virksomheder vil derfor have begge teknologier, de vil i mange tilfælde være at finde i samme produkt. Figur 3 illustrerer en overordnet gruppering af metadata og opgaver for henholdsvis en Metadata-beholder og et register. Som det kan ses, vil der være overlappende funktionalitet, hvorfor det ikke er muligt klart at opdele disse to teknologier.



Figur 3 Opgaver for beholderen og registeret (Kilde: Gartner November 2005)

Uden synlighed, troværdighed og kontrol kan en SOA blive kaotisk og uhåndterbar. Virksomheden skal sikre, at alle deres webservices er baseret på godkendte politikker og kontrakter, samt at de er organiseret, håndterbare og synlige.

En SOA introducerer selv ændringer, der er lige så væsentlige, som de ændringsbehov som en SOA skal adressere. SOA implementeringer skal derfor være bygget til ændringer, for at virksomheden kan opnå behændigheden og besparelserne, som SOA lover. For at kunne genbruge services i en SOA og for at virksomheder kan håndtere ændringer korrekt, er governance afgørende. SOA-governance er kombinationen af politikker, processer og metadata. Vigtigheden af konsekvensanalyser ved ændringer øges i en løst koblet arkitektur.

Metadata er de politikker og beskrivelser af forretningsservices som muliggør identifikation og korrekt brug af disse services. En omfattende mængde information omkring forretningsservicen skal være til stede og forbundet for at understøtte governance- og livscyklus-processer, som skal sikre at virksomhedens SOA forbliver håndterbar.

Virksomheder bør investere i styring af metadata, hvis de vil opnå visionen om løs kobling.