

# Fælles retningslinjer for REST webservices

UDKAST version 0.5

# 2017

# Indhold

---

<b>1. Indledning</b>	<b>3</b>
<b>2. Principper og forretningsbehov for retningslinjer</b>	<b>6</b>
2.1 Serviceudstilling af forretningsobjekter	8
<b>3. Generelle retningslinjer for REST webservices</b>	<b>9</b>
3.1 Anvendelse af Microservice arkitekturstil	9
3.2 Ansvarsfordeling ved kald til webservices	13
3.3 Servicedokumentation	16
3.4 Serviceversionering	21
3.5 Servicelogning	23
3.6 Servicemonitorering	24
3.7 Servicefejlmeddelelser	25
<b>4. Retningslinjer for REST webservices</b>	<b>28</b>
4.1 Modellering af REST webservices	28
4.2 Søgninger i REST webservices	35
4.3 Datarepræsentation i REST webservices	37
4.4 REST webservice specifikation	39
4.5 REST kommunikationsprotokol	40
<b>5. Referenceliste</b>	<b>43</b>

---

# 1. Indledning

Dette dokument beskriver fællesoffentlige retningslinjer for anvendelse af REST webservices i it-løsningerne i den offentlige sektor. Formålet med de fælles retningslinjer for webservices er at skabe bedre interoperabilitet imellem offentlige institutioners it-løsninger.

Retningslinjerne er udsprunget af den fællesoffentlige digitaliseringsstrategi, hvor retningslinjerne er en konkretisering af princip 7 og arkitekturregel 7.1 i Hvidbogen (jf. [HVIDBOG]). Princip 7 tilsiger, at it-løsninger samarbejder effektivt, mens arkitekturregel 7.1 tilsiger, at der skal designes og udstilles snitflader efter fælles integrationsmønstre og tekniske standarder.

En række domæner har i dag allerede domænespecifikke retningslinjer og standarder, fx på sundhedsområdet. Det er væsentligt, at anvendelsen af retningslinjerne finder sted i samspil med eksisterende retningslinjer og standarder, således at der ikke skabes unødigt kompleksitet. De fælles retningslinjer findes særligt anvendelige, hvor der ikke foreligger domænespecifikke standarder og retningslinjer for REST webservices.

Retningslinjerne i dette dokument kan anvendes som en del af den enkelte offentlige institutions metode til skabelse af sammenhængende data, på tværs af it-løsninger inden for egen organisation, hvis organisationen vælger det. Retningslinjerne stiller ikke krav til at offentlige institutioner skal anvende REST over andre arkitekturstile, når data og funktionalitet skal udstilles via webservices.

En webservice er i dette dokument defineret som en udstilling data eller funktionalitet af en serviceudbyder for it-løsninger. En webservice anvendes af én eller flere serviceanvendere. De fælles retningslinjer for REST webservices i dette dokument er en vejledning til serviceudbydere, der skal udstille data og funktionalitet via webservices baseret på REST. Vejledningen skal opfattes som en beskrivelse af god praksis og skal hjælpe serviceudbyderen, når denne designer og udstiller REST webservices rettet imod serviceanvendere fra andre offentlige institutioner.

REST (Representational State Transfer) er en arkitekturstil, der anvendes til udstilling af ressourcer (data) med webservices. REST er baseret på principperne og arkitekturen, som anvendes på internettet, og genbruger få, udbredte standarder og protokoller herfra, f.eks. HTTP.

Serviceanvendere kan have behov for at kalde mere end én webservice. Et eksempel herpå er, hvis en serviceanvender skal foretage en opdatering i to it-løsninger via to forskellige webservices. Sådanne situationer er omfattet af begrebet

”webservice orkestrering”, der inkluderer kald af flere webservices. Retningslinjerne i dette dokument omfatter ikke serviceorkestrering.

Dokumentet er opbygget i fem hovedafsnit.

- Nærværende afsnit 1 udgør en kort indledning.
- Afsnit 2 indeholder en redegørelse for de principper og forretningsbehov, der ligger til grund for de angivne retningslinjers udformning.
- Afsnit 3 indeholder angivelse af generelle retningslinjer, der skal anvendes af REST webservices og kan være anvendelige for alle. Retningslinjerne er grupperet efter tema, således at retningslinjerne hurtigt kan findes ud fra læserens behov.
- Afsnit 4 indeholder en angivelse af retningslinjer, der er specifikke for REST webservices.
- Afsnit 5 indeholder en referenceliste.

Dokumentet henviser til følgende bilag som vedlægges i version 0.8:

- Bilag 1 ”Dokumentation for REST Webservices”
- Bilag 2 ”Fejlstruktur og fejkoder for REST Webservices”
- Bilag 3 ”Nonfunktionelle krav til realisering af retningslinjer”
- Bilag 4 ”Specificering af brug af HTTP til REST”
- Bilag 5 ”Udstilling af forretningsobjekter som REST ressourcer.”

Der er følgende generelle retningslinjer i afsnit 3:

- R01. Udstil minimal funktionalitet i den enkelte webservice
- R02. Målret webservices til konkrete serviceanvendere
- R03. Separér webservices fra konkret implementering
- R04. Separér webservices fra eksterne afhængigheder
- R05. Understøt gentagne forsøg på kald fra serviceanvenderen
- R06. Håndtér dublerede opdaterende kald fra serviceanvendere
- R07. Kræv specifikke informationer ved ansvarsoverdragelse ved kald til webservices
- R08. Dokumentér udstillede webservices i overensstemmelse med den fællesoffentlige dokumentationsramme for webservices
- R09. Opmærk webservices i henhold til fællesoffentlige emnesystematikker
- R10. Opmærk webservices med følsomhed eller fortrolighed af data
- R11. Dokumentér servicespecifikke fejlkoder for webservices
- R12. Dokumentér drift af forskellige versioner af webservices
- R13. Anvend semantisk versionering af webservices
- R14. Udstil gamle og nye versioner når en webservice ændres
- R15. Log alle væsentlige kald til webservices
- R16. Anvend transaktionsidentifikatorer ved kald og svar
- R17. Understøt monitorering af udstillede webservices
- R18. Returnér servicespecifikke fejl som standardiserede fejlmeddelelser

Der er følgende specifikke retningslinjer i afsnit 4:

- R19. Udstil webservices som REST ressourcer
- R20. Udstil forretningsprocesser som REST ressourcer
- R21. Modellér REST ressourcer med udgangspunkt i forretningsmodellering
- R22. Navngiv REST ressourcer ud fra forretningsmodellens begreber
- R23. Navngiv REST ressourcer ud fra REST arkitekturstilen
- R24. Udstillede REST ressourcer har unikke, sikre identifikatorer
- R25. Udstil REST ressourcers relaterede entiteter
- R26. Returnér REST ressourcer med hypertext links
- R27. Anvend standardiserede REST fremsøgningsparametre
- R28. Understøt søgning med delresultater
- R29. Repræsenter REST ressourcer i et standardiseret dataformat
- R30. Deklarér REST ressourcer datarepræsentationer
- R31. Overfør tekst i en internationaliseret repræsentation
- R32. Dokumentér REST webservices via et standardiseret skema
- R33. Anvend HTTP som fællesoffentlig REST kommunikationsprotokol
- R34. Anvend HTTPs mekanismer til effektiv kommunikation
- R35. Anvend HTTP sikkert til REST ressourcer

## 2. Principper og forretningsbehov for retningslinjer

Retningslinjerne for REST webservices er udarbejdet med henblik på at være bredt anvendelige i den offentlige sektor, baseret på praktisk erfaring samt være anvendelige i realiseringen af den fællesoffentlige digitaliseringsstrategi. For at sikre dette, er der i indeværende afsnit 2 specificeret tre overordnede principper samt en række forretningsbehov, som retningslinjerne opfylder. Disse principper og forretningsbehov har således udgjort grundlaget for udarbejdelsen af retningslinjerne, der fremgår af afsnit 3 og 4.

Principperne og forretningsbehovene for retningslinjerne er dokumenteret i indeværende afsnit 2 med henblik på at sikre læserens forståelse for, hvorfor netop de i afsnit 3 og 4 specificerede retningslinjer er valgt til at sikre interoperabilitet mellem it-løsninger på tværs af offentlige institutioner.

Den offentlige institution, som ønsker at anvende retningslinjerne bør undersøge om organisationen har væsentligt andre forretningsbehov. Endvidere er forretningsbehovene ikke altid samstemmende, fx kan det være en udfordring at have en snitflade som er nem for serviceanvendere og serviceudbydere. Retningslinjerne indeholder således en afvejning hvorledes forretningsbehovene samlet set bedst kan opfyldes.

Følgende tre principper gælder for alle retningslinjerne:

- Retningslinjerne skal kunne anvendes bredt af offentlige institutioner
- Retningslinjerne skal baseres på viden fra eksisterende anvendelser af og retningslinjer for webservices, bl.a. fra SKAT, Sundhedsdatastyrelsen, Grunddataprogrammet, KOMBIT.
- Retningslinjerne skal være langtidsholdbare, men samtidig være tilstrækkeligt konkrete og implementeringsnære til, at retningslinjerne kan inddrages af initiativerne i digitaliseringsstrategien.

Forretningsbehov for REST webservices er grupperet i tre temaer.

Første tema er ”Forretningsbehov vedrørende serviceanvendere og serviceudbydere” som adresserer dilemmaet i at designe services, der er bedst for både serviceanvendere og serviceudbydere. I dette dokument ønskes der fokus på, at serviceudbydere har et udefra og ind perspektiv, når REST webservices skal designes.

Andet tema er ”Forretningsbehov vedrørende data og funktionalitet”, hvor der fokuseres på behovene til REST webservices udstilling af data sker på en måde, som sikrer mest mulig interoperabilitet.

Tredje tema omhandler behovene for at REST webservices kan indgå i et fællesoffentligt it-landskab på en forvaltningsmæssig sikker og effektiv måde.

### **Forretningsbehov vedrørende serviceanvendere og serviceudbydere:**

- F1. REST webservices skal være omkostningseffektive set i et fællesoffentligt perspektiv, således at der fokus på totalomkostningen for serviceanvendere og serviceudbydere i webservicens levetid. REST webservices skal kunne realiseres med eksisterende og velbeskrevne metoder og udbredte værktøjer og rammeværk af serviceudbydere, der ønsker at udstille data i en webservice.
- F2. Design af REST webservices skal tage udgangspunkt i serviceanvenderens behov. Det vil sige, at REST webservices udstilling af data og funktionalitet er intuitiv og enkel at anvende for en serviceanvender.
- F3. Ændringer i REST webservices, skal udformes, således at det er så omkostningseffektivt som muligt for serviceanvendere. Ændring af udstillede webservices medfører nødvendige tilpasninger hos serviceanvendere. Ændringer hos serviceanvendere medfører ekstra omkostninger og kan give stærke bindinger i it-landskabet ved, at en serviceudbyder ikke kan ændre en webservice før væsentlige serviceanvendere eller i værste tilfælde alle serviceanvendere er klar. Hvis antallet af serviceanvendere er højt, udgør omkostninger til serviceanvenderes it-løsninger typisk størstedelen af den samlede omkostning ved en ændring af REST webservice. Design af REST webservices skal derfor have som mål at minimere konsekvenser for serviceanvendere ved ændringer i en webservice, da det vil være totaløkonomisk fordelagtigt og skabe færre bindinger som sænker evnen til forandring i it-landskabet.
- F4. Webservices skal have en entydig mekanisme til ansvarsoverdragelse mellem myndigheder, således at der altid er klarhed over, hvilken myndighed der har ansvaret for en given myndighedsopgave. Ansvarsoverdragelse kan fx være frigørende virkning for en myndighed ved aflevering af meddelelser til Digital Post.
- F5. REST webservices skal kunne understøtte forandringer i organisationers forretningskrav ved hurtigt at kunne understøtte ændringer i data og funktionalitet, samt hurtigt kunne skalere til ændret forbrug.

### **Forretningsbehov vedrørende data og funktionalitet:**

- F6. REST webservices skal udstille forretningsobjekter som ressourcer på ensartet måde for at sikre interoperabilitet, jf. afsnit 2.1
- F7. Udstilling af REST webservices skal omfatte dokumentation af webservices, herunder en dokumentation af regler for syntaks, semantik og en beskrivelse af snitfladen.
- F8. REST webservices skal anvende eksisterende og kommende fællesoffentlige datastandarder og profiler for REST.
- F9. REST webservices skal udstille data i JSON eller XML.

F10. REST webservices skal tage højde for, at data kan have forskellige grader af fortrolighed, hvilket stiller forskellige krav<sup>1</sup> til egenskaberne ved en webservice. Fx kan en webservice udstille personfølsomme data.

#### **Forretningsbehov vedrørende stabil og sikker driftsafvikling:**

- F11. REST webservices skal kunne dokumentere anvendelsen af data og funktionalitet, herunder svartider og fejlsituationer.
- F12. REST webservices skal kunne overvåges og forvaltes på tværs af webservices, ved at REST webservices skal udstille tilstrækkelige data til at kunne indgå i orkestrering, herunder indeholde dubletkontrol og ensartet struktur af fejlbeskeder.
- F13. REST webservices skal kunne udstille adgang til data og funktionalitet af fortrolig eller følsom karakter. Når serviceudbyderen anvender (f.eks. danske udgaver af) udbredte, gældende sikkerhedsstandarder i stedet for at definere egne sikkerhedsmodeller, vil kravene til databeskyttelse med højere sandsynlighed være opfyldt.

## 2.1 Serviceudstilling af forretningsobjekter

Serviceudbydere udstiller data til serviceanvendere via webservices. Serviceanvendere har typisk behov for adgang til de udstillede webservices i forbindelse med gennemførelse af en forretningsproces. Forretningsprocesser vil løbende ændre sig, f.eks. på baggrund af øget eller ændret digitalisering, ændret lovgivning eller modernisering af it-løsningerne. Tilsvarende vil snitfladen ændre sig i samme takt.

Ved design af en webservice skal det fastlægges, hvorledes data udstilles med en passende granularitet. Ved granularitet forstås i denne sammenhæng, hvor megen funktionalitet eller hvor meget data, der udstilles af én konkret webservice eller service operation.

---

<sup>1</sup> For eksempel er der forskel på krav til webservices, der udstiller offentligt tilgængelige data, og webservices der udveksler personfølsomme data mellem myndigheder.



## 3. Generelle retningslinjer for REST webservices

Indeværende afsnit indeholder retningslinjer, der understøtter forretningsbehovene beskrevet i afsnit 2 og er generelt anvendelige ved implementering af webservices til it-løsninger i den offentlige sektor. Disse retningslinjer er således ikke blot anvendelige til REST webservices, der er fokus for indeværende dokument, men også webservices baseret på eksempelvis SOAP. For de retningslinjer, der er specifikke for REST webservices, henvises til afsnit 4.

Retningslinjerne beskrives efter følgende mønster:

Navn	Angiver nummeret og navnet på retningslinjen
Retningslinje	Beskriver retningslinjen klart og præcist
Rationale	Beskriver forretningsværdien ved at følge retningslinjen
Understøttede forretningsbehov	Beskriver de konkrete forretningsbehov, som retningslinjen understøtter
Implikation	Beskriver hvilken indvirkning retningslinjen har på forretningsmæssig og teknisk implementering

Før beskrivelsen af hver retningslinje er der en kort indledning, der forklarer baggrunden for retningslinjen. Denne indledning anses ikke som en del af retningslinjen, men skal blot give læseren yderligere viden, kontekst og baggrund for at forstå den pågældende retningslinje.

De generelle retningslinjer for webservices er opdelt i følgende områder:

- Anvendelse af microservice arkitekturstil
- Ansvarsfordeling ved kald til webservices
- Servicedokumentation
- Serviceversionering
- Servicelogning
- Servicemonitorering
- Servicefejlmeddelelser

Retningslinjerne inden for hvert af disse områder beskrives i det følgende.

### 3.1 Anvendelse af Microservice arkitekturstil

Der findes flere strategier og arkitekturstilarter til design af webservices, således at konsekvenser for serviceanvendere reduceres. ”Microservices” er en arkitek-

turstil, der fordrer, at webservices designes med fokus på forretningsvendt og afgrænset funktionalitet, jf. [MICROSERVICES]. Retningslinjerne i dette afsnit tager udgangspunkt i denne arkitekturstil.

En konsekvens af at følge Microservice-arkitekturstilen er, at webservices er så små som muligt. De har så minimal funktionalitet som muligt, men leverer fortsat forretningsværdi. En anden konsekvens af denne arkitekturstil er, at serviceudbyderen typisk udstiller et væsentligt større antal webservices, da hver webservice ikke er generisk, men specifik. Til gengæld er der færre serviceanvendere, der påvirkes, når specifikke webservices skal ændres.

Microservice arkitekturstilen hjælper dermed serviceudbyderen ved at øge dennes forandringsevne, så det er nemmere at videreudvikle webservices i takt med forretnings behov.

Navn:	R01. Udstil minimal funktionalitet i den enkelte webservice
<b>Retningslinje</b>	Webservices skal designes med adgang til data eller funktionalitet svarende til netop én (1) forretningsmæssig kapabilitet.
<b>Rationale</b>	Ændringer i den enkelte forretningsmæssige kapabilitet sker mindre ofte, sammenlignet med det samlede spektrum af forretningskapabiliteter. Ændringer til én forretningskapabilitet bør i princippet kun medføre ændringer i den eller de webservices, der udstiller kapabiliteten. På denne måde minimeres behovet for ændringer i etablerede webservices, når forretningskapabiliteter ændres.
<b>Understøttede forretningsbehov</b>	Retningslinjen understøtter forretningsbehov F2 og F3, idet den beskytter serviceanvendere imod ændringer.
<b>Implikation</b>	Den enkelte webservice bør kun udstille et enkelt forretningsobjekt eller en enkelt, sammenhængende funktionalitet ad gangen. En webservice løser ideelt kun én (1) opgave <sup>2</sup> . En konsekvens af denne minimering er, at genbrugeligheden af den enkelte webservice bliver reduceret. Til gengæld vil det samlede økosystem omkring den samlede mængde af webservices bedre være i stand til at håndtere ændringer. Eksempelvist udstilles i OIO standarderne for sags og dokumentområdet en ”Ret” funktion for hvert forretningsobjekt, herunder objektet ”sag”. Funktionen ”Ret sag” kunne således understøtte redigering af enten attributter, relationer eller tilstande. Det medfører at samtlige serviceanvendere skal anvende ”Ret sag” (da denne genbruges). En mere fint granuleret tilgang med f.eks. at udstille én ressource til attributter og

<sup>2</sup> Svarende til Single Responsibility princippet fra SOLID ([https://en.wikipedia.org/wiki/SOLID\\_\(object-oriented\\_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))).

	en anden ressource per relation, vil kræve flere udstillede ressourcer, men det vil sænke kompleksiteten og antallet af bindinger for serviceanvendere.
--	---

En webservice som udstiller generiske operationer<sup>3</sup> kan anvendes af mange serviceanvendere med forskellige behov. Dermed vil en sådan generisk webservice skabe mange bindinger af snitfladen. En ændring af en generisk webservice kan således medføre ændringer til serviceanvendere, selvom ændringen ikke giver værdi for serviceanvenderen. Serviceudbyderen har dermed væsentligt reducerede muligheder for at ændre generiske webservices uden meromkostninger eller risici. En af de bærende ideer bag Microservice-arkitekturstilen er, at webservices designes så deres funktionalitet er rettet mod (konkrete) serviceanvendere eller konkret anvendelse ("Consumer driven"). Webservices designes dermed ikke generisk, uden at målrette til den faktiske anvendelse ved at forsøge at dække forventede behov fra mange evt. påtænkte serviceanvendere.

En konsekvens af denne målretning kan være, at serviceudbyderen skal udstille flere forretningsvendte webservices med mindre funktionalitet, frem for én større generisk webservice med meget funktionalitet. De mange forretningsvendte webservices supplerer hinanden, og gør det mindre omkostningstungt samlet set, når den enkelte webservice skal forandres.

Navn:	R02. Målret webservices til konkrete serviceanvendere
<b>Retningslinje</b>	Webservices skal designes med den funktionalitet, konkrete serviceanvendere har behov for.
<b>Rationale</b>	Hvis webservices designes så deres udstillede funktionalitet målrettes bestemte serviceanvendere eller konkret anvendelse, i stedet for at være så generiske som muligt, vil senere ændringer påvirke færrest mulige serviceanvendere.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 og F3 ved at målrette webservicen, og dermed beskytte flertallet af serviceanvendere imod ændringer. Retningslinjen afviger fra OIO serviceprincippet "Services kan genbruges" (se [OIOSVC]), idet genbrug vurderes af lavere prioritet end målretning til konkrete anvendere og beskyttelse af anvenderne imod ændringer.

---

<sup>3</sup> Et eksempel på en generisk webservice er en webservice der tillader generel søgning i en mængde udstillede forretningsobjekter, hvis indre repræsentation (attributter og relationer) direkte gøres tilgængelig for serviceanvendere.

<b>Implikation</b>	<p>Ved webservice design bør målgruppen for den enkelte webservice afdækkes. Webservices bør designes målrettet de medlemmer af den samlede målgruppe, der har de samme behov, i stedet for at designe en generisk webservice, der også forsøger at tilfredsstille uidentificerede serviceanvenderes behov. Eksempelvist kan redigering af en sag medføre en opdatering af attributter og tilføjelse af journalnotater. Hvis redigering af en sag er en generisk webservice, vil webservicen ændre sig hver gang, der kommer ændringer i datamodellen. En serviceanvender har måske kun behov for at opdatere journalnotater eller kun attributter, og kan dermed nøjes med en enklere funktion.</p> <p>En mulig konsekvens af retningslinjen er, at der skal udstilles næsten ens varianter af webservices.</p>
--------------------	--

En anden bærende ide i forbindelse med Microservice-arkitekturstilen er at afgrænse servicens eksterne afhængigheder og kontekst ("Bounded context").

I stedet for at udstille en stor og kompleks model, der indeholder alle egne forretningsobjekter såvel som forretningsobjekter fra andre domæner i forbindelse med en webservice, afgrænser webservicen de udstillede data.

Navn:	R03. Separér webservices fra konkret implementering
<b>Retningslinje</b>	Webservices skal designes, så de afkobler implementeringen af interne og eksterne forretningsobjekter fra den snitflade, der udstilles imod serviceanvenderne.
<b>Rationale</b>	Ved at anvende en facade, der afskærmer serviceanvenderen fra den konkrete implementering (i kildekode) af webservicen, vil serviceudbyderen opnå større friheder i implementeringen af webservicen.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F3 ved at afskærme serviceanvenderen fra den konkrete implementering af webservicen. Retningslinjen supplerer OIO serviceprincippet "Services indkapsler data/funktionalitet" (se [OIOSVC]).
<b>Implikation</b>	Ved servicedesign bør det afdækkes, netop hvilke dele af et forretningsobjekt, der er behov for at gøre tilgængeligt for anvendere. Webservicens design bør give disse dele en repræsentation, der ikke fastholder serviceudbyderen i den samme konkrete implementering af webservicen i hele servicens levetid.

Webservicen udstiller kun de data, serviceanvenderen faktisk har brug for, og udgør en facade foran forretningsobjekter fra andre domæner. Webservicen definerer med andre ord et eget forretningsobjekt, der skygger for det eksterne. Hvis

serviceanvenderen har brug for mere information om det andet domænes forretningsobjekter, skal det andet domænes webservices anvendes.

<b>Navn:</b>	<b>R04. Separér webservices fra eksterne afhængigheder</b>
<b>Retningslinje</b>	Webservices skal designes, så snitfladen skal afgrænse sig til kun at bruge egne domæneobjekter, og ikke have eksterne afhængigheder til andre domæners objekter.
<b>Rationale</b>	Ved at anvende en facade foran eksterne forretningsobjekter, får serviceanvenderen kun et smalt perspektiv af det fremmede forretningsobjekt, for at reducere kompleksitet og gardere imod konsekvenser af ændringer.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F3 ved at afskærme serviceanvenderen fra fremmede forretningsobjekter. Retningslinjen supplerer OIO serviceprincippet ”Services indkapsler data/funktionalitet” (se [OIOSVC]).
<b>Implikation</b>	Ved servicedesign bør det afdækkes, netop hvilke dele af et fremmed forretningsobjekt, der er behov for at gøre tilgængeligt for anvendere.

### 3.2 Ansvarsfordeling ved kald til webservices

Ved integrationer mellem systemer er det centralt, at ansvaret for gennemførelse af integrationens funktioner er tydeligt placeret. Dette gør sig særligt gældende ved kald til webservices der opdaterer data, eller ved overdragelse af ansvar, f.eks. fra én myndighed eller aktør til en anden.

Det er vigtigt at sikre, at opdateringer gennemføres fuldstændigt og netop én gang. Serviceudbyderen skal derfor kunne antage, at serviceanvenderen kan gennemføre alle kald<sup>4</sup> til en webservice, indtil serviceanvenderen har modtaget et svar<sup>5</sup> fra webservicen.

Det er kun serviceanvenderen, der kan vide med sikkerhed om et kald til en webservice er gennemført, nemlig når et svar på kaldet modtages fra webservicen. Webservicen ”selv” kan ikke ”vide” om svaret er modtaget af serviceanvenderen, idet svaret kan gå tabt undervejs.

**Navn: R05. Understøt gentagne forsøg på kald fra serviceanvenderen**

<sup>4</sup> Ved et kald forstås i denne forbindelse, at serviceanvenderen afsender en forespørgsel, f.eks. udformet som et HTTP kald med headere og data, jf. HTTP protokollen.

<sup>5</sup> Ved et svar forstås i denne forbindelse, at en webservice returnerer informationer som konsekvens af modtagelsen af et kald. Informationerne kan f.eks. være et HTTP svar med headere og data, jf. HTTP protokollen.

<b>Retningslinje</b>	Serviceanvenderen har ansvar for fejlhåndtering ved kald til en webservice og webservicen skal derfor returnere sigende fejlmeddelelser, hvis fejlen opstår i webservicen, så serviceanvenderen om nødvendigt kan genforsøge kaldet indtil kaldet kan gennemføres.
<b>Rationale</b>	En klar ansvarsplacering i forbindelse med gennemførelse af kald er nødvendig for, at informationer og evt. ansvar for behandling overdrages korrekt. Det er kun serviceanvenderen, der kan afgøre om svaret på et kald er modtaget.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F4.
<b>Implikation</b>	Det er serviceanvenderen, der er ansvarlig for at gennemføre et kald, herunder at genforsøge kaldet, indtil et svar (en fejl eller en kvittering) modtages fra webservicen. Serviceudbyderen skal designe webservices, således at et konkret kald kan genforsøges af serviceanvenderen én eller flere gange, uafhængigt af om der opstår fejl under eller i forbindelse med kaldet eller svaret (se også retningslinje R06). Hvis webservices kræver at en serviceanvender medsender en transaktionsidentifikator ved alle kald, jf. retningslinje R16, kan transaktionsidentifikatoren anvendes til at detektere gentagne forsøg, og håndtere gentagelsen korrekt.

Når serviceanvendere foretager kald til opdaterende<sup>6</sup> webservices, kan de jf. retningslinje R05 ovenfor være nødt til at ”afsende” opdateringen flere gange for at sikre, at opdateringen er gennemført.

Det er kun webservicen, der kan afgøre om den opdatering faktisk er gennemført. Webservicen skal derfor beskytte serviceanvendere imod effekterne af dublerede opdateringer. Ved ”effekten” forstås i denne forbindelse, at opdateringen for nogle webservices’ vedkommende blot kan gentages (hvis opdateringen kun overskriver data). Andre typer opdateringer – f.eks. tilføjelser – har en natur, hvor data-opdateringen nødvendigvis skal ske netop én gang. I så fald skal webservicen sikre dette.

---

<sup>6</sup> Ved ”opdaterende” forstås i denne forbindelse et kald der foretager en ændring til et forretningsobjekt, f.eks. oprettelse, ændring, eller sletning af data i eller et helt forretningsobjekt. Den medhørende logning anses ikke som en opdatering.

Uanset hvordan webservicen håndterer opdateringen, skal serviceanvenderen have et svar som om opdateringen er udført netop én gang. En webservice skal med andre ord være ”idempotent”<sup>7</sup>.

Navn:	R06. Håndtér dublerede opdaterende kald fra serviceanvendere
<b>Retningslinje</b>	Webservices skal kunne modtage gentagne opdaterende kald <sup>8</sup> fra serviceanvendere, og sikre at kaldet behandles så opdateringen har effekt, som om kaldet er udført netop en gang.
<b>Rationale</b>	Serviceudbyderen skal sikre sig imod eventuelle problemer ved gentagne opdateringsforsøg, da serviceanvenderen ikke er i stand til at forhindre dette.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1 og F4. Retningslinjen er relateret til OIO integrationsmønsteret ”Dokumentforsendelse” (se [OIOMØNSTRE]).
<b>Implikation</b>	Serviceudbyderen er ansvarlig for, at der ingen utilsigtede konsekvenser er ved gentagne - korrekte - forsøg fra serviceanvenderens side på opdateringer. Hvis webservices kræver, at en serviceanvender medsender en transaktionsidentifikator ved alle kald, jf. retningslinje R16, kan transaktionsidentifikatoren anvendes til at detektere gentagne forsøg og håndtere gentagelsen korrekt. Afhængigt af den konkrete webservice kan opdateringen i nogle tilfælde blot gentages ved hvert kald. Logning hos serviceudbyderen er i denne forbindelse ikke at betragte som en opdatering. Det er således muligt, at det af loggen hos serviceudbyderen fremgår, at serviceanvender har genforsøgt et kald.

Når der skal ske overdragelse af ansvar for en myndighedsopgave mellem to parter – f.eks. to myndigheder, eller en myndighed og en privat virksomhed – er en kvittering om, at de nødvendige data er overført (en ”teknisk kvittering”) som regel utilstrækkelig. Der er i sådanne situationer brug for, at den modtagende part eksplicit kvitterer for, at opgaven som minimum er modtaget, forstået og vil blive behandlet (en ”forretningskvittering”).

Navn:	R07. Kræv specifikke informationer ved ansvarsoverdragelse ved kald til webservices
-------	---

<sup>7</sup> Idempotent for REST webservices er kort beskrevet her: <http://restcookbook.com/HTTP%20Methods/idempotency/>

<sup>8</sup> Ved ”opdateringskald” forstås i denne forbindelse et kald der foretager en ændring til et forretningsobjekt, f.eks. oprettelse, ændring, eller sletning af data i eller et helt forretningsobjekt.

<b>Retningslinje</b>	Webservices, der understøtter overdragelse af myndighedsansvar, skal kræve at specifikke overdragelsesinformationer inkluderes i kaldet, og skal eksplicit returnere en kvittering for, at overdragelsen er accepteret.
<b>Rationale</b>	Det skal være uomtvisteligt, hvornår en myndighed eller en organisation (som serviceanvender) har overdraget et ansvar til en anden myndighed eller en organisation (som serviceudbyder). For at sikre dette, skal webservicen kræve at serviceanvenderen har inkluderet specifikke informationer om overdragelse af ansvar i kaldet. Den valgte kommunikationsprotokols tekniske kvitteringer (http) kan ikke nødvendigvis erstatte forretningsmæssige kvitteringer for ansvarsoverdragelse. Fx kan en snitflade måske ikke synkront opdatere de bagvedliggende forretningssystemer. Dermed kan serviceudstiller ikke give en forretningsmæssig kvittering før alle bagvedliggende systemer er opdateret korrekt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F4.
<b>Implikation</b>	<p>Serviceudbyderen skal designe webservicen således, at eksplícitte overdragelsesinformationer skal være til stede i et korrekt kald til webservicen. Disse informationer kan f.eks. omfatte:</p> <ul style="list-style-type: none"> <li>• Identifikation af den ansvarsafgivende organisation/myndighed</li> <li>• Identifikation af den ansvarsmodtagende organisation/myndighed</li> <li>• En unik identifikator af denne ansvarsoverdragelse</li> <li>• En identifikation af hvilket ansvar, der overdrages, f.eks. ved reference til en aftalt klassifikation, emnesystematik eller lovgrundlag.</li> </ul> <p>Serviceudbyderen skal realisere webservicen, så den returnerer et kvitteringssvar, der indikerer, at ansvaret er overdraget. Kvitteringssvaret kan f.eks. bestå af overdragelsesinformationerne (se ovenfor) samt en status for overdragelsen (f.eks. accepteret eller afvist).</p> <p>Serviceudbyderen er ansvarlig for at sikre behørig logning af modtagelse af overdragelsen og dens udfald.</p>

### 3.3 Servicedokumentation

Serviceudbydere skal dokumentere alle udstillede webservices. Dokumentationen skal målrettes serviceanvenderens behov for information, både forretningsmæssigt og teknisk. God dokumentation målrettet serviceanvendere kan reducere omkostningerne, når udstillede webservices skal anvendes, da det bliver lettere for serviceanvenderen at vælge og bruge udstillede webservices korrekt.



Serviceanvenderes behov består både af forretningsforståelse af de data, der udveksles via webservicen, den tekniske forståelse for, hvordan og i hvilken repræsentation data udveksles, samt koblingen mellem forretningsforståelse og konkrete datarepræsentationer. Specifikationen af den udstillede webservice er en central del af dokumentationen.

Dokumentationen for webservices skal derfor følge den fællesoffentlige dokumentationsramme for webservices. Dokumentationsrammen publiceres og vedligeholdes på Digitaliseringsstyrelsens hjemmeside, og udgør et bilag til retningslinjerne i dette dokument jf. bilag 1 ”Dokumentation for REST Webservices”.

God dokumentation er opdelt og opmærket på en måde, der gør, at dokumentationens enkeltdele målrettes de forskellige roller hos serviceanvenderen (f.eks. forretningsansvarlig eller it-arkitekt).

Det er en nødvendig forudsætning, at den udstillede dokumentation er ajourført, så den svarer til den udstillede webservice. Dokumentation, der ikke er ajourført, har meget begrænset værdi hos serviceanvendere.

Navn:	<b>R08. Dokumentér udstillede webservices i overensstemmelse med den fællesoffentlige dokumentationsramme for webservices</b>
<b>Retningslinje</b>	Alle webservices skal være dokumenterede i overensstemmelse med den fællesoffentlige dokumentationsramme, og dokumentationen skal være let tilgængelig, ajourført og svare til de udstillede webservices.
<b>Rationale</b>	Det skal være muligt for serviceanvendere at forstå, hvordan udstillede webservices skal anvendes uden at skulle konsultere eller kontakte serviceudbyderen. Ved f.eks. at synliggøre transformationen mellem forretningsobjekter og webservicen sikres synlighed og ensartethed, så ressourcen har samme betydning for både serviceudbyder og serviceanvender. Let tilgængelig dokumentation medfører, at serviceanvenderen har let ved at finde de relevante dele af dokumentationen, når denne skal anvende webservicen.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 Retningslinjen supplerer OIO serviceprincippet ”Services publiceres” (se [OIOSVC]).
<b>Implikation</b>	Den enkelte webservice skal være dokumenteret, herunder hvordan den er modelleret i forhold til de bagvedliggende forretningsobjekter. Krav til dokumentationen fremgår af den fællesoffentlige dokumentationsramme for webservices, der er et bilag til retningslinjerne i dette dokument. Dokumentationens enkeltdele opmærkes, således at den bliver lettere tilgængelig for forskellige målgrupper, f.eks. udviklere, arkitekter og forretningspecialister. Opmærkningen skal

	<p>følge angivelserne i den fællesoffentlige dokumentationsramme for webservices.</p> <p>Ved ændringer til webservicen skal dokumentationen opdateres senest når ændringen idriftsættes, således at en webservice ikke er udstillet med forældet dokumentation.</p>
--	---

Serviceudbyderen skal opmærke webservices i henhold til emnesystematik, således at serviceanvenderen tydeligt kan se hvilken offentlig forvaltning, forvaltningsproces eller anden proces hos serviceudbyderen, webservicen er udstillet i relation til.

<b>Navn:</b>	<b>R09. Opmærk webservices i henhold til fællesoffentlige emnesystematikker</b>
<b>Retningslinje</b>	Serviceudbyderen skal opmærke udstillede webservices i henhold til fællesoffentlige emnesystematikker.
<b>Rationale</b>	Som en del af den forretningsrelaterede del af dokumentation for webservices skal serviceanvenderen bibringes viden om, hvilken offentlig forvaltning, forvaltningsprocesser eller andre processer, webservicen er en del af hos serviceudbyderen.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 Retningslinjen supplerer OIO serviceprincippet ”Services publiceres” (se [OIOSVC]).
<b>Implikation</b>	Dokumentationen af den enkelte webservice skal inkludere opmærkning i henhold til fællesoffentlige emnesystematikker, dvs. enten KLE eller FORM. Opmærkningen påføres den enkelte webservice. Hvis en webservice indgår i flere forvaltningsprocesser eller forvaltnings-emner, skal alle emnerne angives i dokumentationen. Desuden angives en vejledning til serviceanvenderen om, i hvilke tilfælde de forskellige emner bliver aktuelle i forhold til kald af webservicen.

Serviceudbyderen skal opmærke webservices med angivelse af følsomhed for de data, der overføres ved brug af webservicen (dvs. både kald og svar). Opmærkningen angives ikke i selve kaldet eller svaret, men som en del af webservicens dokumentation.

<b>Navn:</b>	<b>R10. Opmærk webservices med følsomhed eller fortrolighed af data</b>
<b>Retningslinje</b>	Serviceudbyderen skal opmærke udstillede webservices med følsomhed eller fortrolighed af de data, der skal angives ved kald, og de data, der returneres ved svar.
<b>Rationale</b>	Som en del af den sikkerhedsrelevante del af dokumentation for webservices skal serviceudbyderen angive følsomhed eller

	fortroligheden, så det fremstår tydeligt overfor serviceanvenderen, hvordan data til og fra webservice skal behandles.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 Retningslinjen supplerer OIO serviceprincippet ”Services publiceres” (se [OIOSVC]).
<b>Implikation</b>	Dokumentationen af den enkelte webservice skal inkludere opmærkning i henhold klassificering af følsomhed og fortrolighed i den fællesoffentlige dokumentationsramme for webservices. Opmærkningen påføres den enkelte webservice i form af angivelse af fortroligheds- eller følsomhedsniveauer fra en klassifikation af niveauer i den fællesoffentlige dokumentationsramme. Hvis en webservice har flere niveauer, angives det højeste niveau af følsomhed eller fortrolighed, jf. klassifikation af niveauer i den fællesoffentlige dokumentationsramme.

Serviceanvenderen har behov for at kende til de servicespecifikke fejl, en webservice kan returnere som svar på kald til webservicen. Servicespecifikke fejl er ikke de fejl, der opstår i infrastrukturen mellem serviceanvender og webservice. Et eksempel på en servicespecifik fejl er en fejl, der opstår under validering af kaldet fra serviceanvenderen, fordi serviceanvenderen har angivet en værdi, der ikke tillades af webservicen. Denne fejl er servicespecifik, da den (først) opstod i forbindelse med webservicens behandling af kaldet, efter at kaldet var transporteret til webservicen.

<b>Navn:</b>	<b>R11. Dokumentér servicespecifikke fejlkode for webservices</b>
<b>Retningslinje</b>	Alle servicespecifikke fejl, der kan returneres fra en webservice, skal dokumenteres, og dokumentationen skal indgå i webservicens dokumentation og gøres tilgængelig for serviceanvenderne.
<b>Rationale</b>	Serviceanvenderne skal kunne forstå, hvorfor kald til en udstillet webservice ikke kan gennemføres, så servicenanvenderen eventuelt kan kompensere for problemet. Hvis servicespecifikke fejl dokumenteres, gives serviceanvenderen mulighed for selv at løse problemerne, i stedet for at skulle bebyrde serviceudbyderen med bistand til at løse problemet.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 Retningslinjen supplerer OIO serviceprincippet ”Services publiceres” (se [OIOSVC]).
<b>Implikation</b>	Dokumentationen for webservices skal inkludere en oversigt over alle servicespecifikke fejl, samt dokumentation af hver enkelt servicespecifik fejl. Dokumentationen af den enkelte servicespecifikke fejl bør angive hvilken fejlkode (en værdi), fejlen er identificeret ved,

	<p>en fejlbeskrivelse og årsag, samt serviceudbyderens forslag til hvordan serviceanvenderen kan løse problemet.</p> <p>Dokumentationen af servicespecifikke fejl er en del af webservices dokumentation, og skal udstilles til serviceanvenderne.</p>
--	--

Serviceudbydere skal tydeliggøre ændringer til webservices, så serviceanvenderne kan afgøre om ændringens effekt fordrer konsekvensændringer hos dem selv. Serviceudbydere skal dokumentere deres strategi for, i hvilket omfang forskellige versioner af en udstillet webservice driftsafvikles samtidigt. Serviceanvenderne bringes dermed en forståelse for, hvor hurtigt de skal tilpasse sig ved ændringer til webservice.

Navn:	R12. Dokumentér drift af forskellige versioner af webservices
<b>Retningslinje</b>	Serviceudbyderen skal dokumentere vilkårene i forbindelse med samtidig drift af inkompatible versioner af den samme webservice.
<b>Rationale</b>	Serviceanvenderne skal tydeligt være informeret om de vilkår, der gælder, når serviceudbyderen udstiller forskellige versioner af en webservice. Serviceanvenderen skal vide, hvor lang tid, denne har til at tilpasse sig en ny version af en webservice, hvis der foretages en ikke-kompatibel ændring af denne (se også retningslinje R13).
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2
<b>Implikation</b>	<p>Serviceudbyderen skal fastlægge og dokumentere sin release- og versioneringsstrategi (se også R13), og publicere denne sammen med dokumentationen for udstillede webservices. Det bør af dokumentationen fremgå:</p> <ul style="list-style-type: none"> <li>• Hvordan webservices versioneres (se også R13)</li> <li>• Hvor længe gensidigt inkompatible webservices udstilles i flere udgaver, samt antallet heraf</li> <li>• Om og i givet fald hvor længe i forvejen serviceudbyderen adviserer serviceanvenderne om ændringer</li> <li>• Hvordan serviceanvenderne eventuelt kan give sig til kende overfor serviceudbyder</li> </ul> <p>Serviceudbyderen skal udstille en oversigt over konkrete datoer for publicering af nye webservices samt ændringer til og nedlæggelse af eksisterende webservices.</p>

### 3.4 Serviceversionering

Det er centralt for serviceanvendere, at de er i stand til at afgøre om en ændring til en webservice kræver korresponderende ændringer hos serviceanvenderen. Konsekvenserne af sådanne ændringer skal generelt minimeres.

Når ændringer er nødvendige, skal webservicens version tydeligt indikere overfor serviceanvenderen, om webservicen er kompatibel med den tidligere udgave. Ved kompatibilitet forstås i denne forbindelse at én version af en webservice er kompatibel med en tidligere version, hvis der ingen ændringer kræves hos serviceanvenderen.

Versionering af udstillede webservices er via mekanismen ”semantisk versionering”<sup>9</sup> et middel til at gøre det klart og tydeligt overfor serviceanvenderen, om der er fortaget ændringer, der kræver handling fra servicesanvenderens side.

Ifølge reglerne for semantisk versionering anvendes et versionsnummer med tre dele: major version, minor version og patch version. Reglerne for versionsnumre er således:

- To versioner af en webservice med forskellige major versioner (f.eks. 1.1 og 2.0) er ikke kompatible. En serviceanvender, der anvender version 1.1 af webservicen, vil skulle foretage væsentlige ændringer for at kunne anvende version 2.0 af webservicen.
- Når to versioner af en webservice har samme major version men forskellige minor versioner (f.eks. 1.1 og 1.2), er version 1.2 af webservicen kompatibel med version 1.1. Serviceanvenderen skal derfor ikke foretage sig noget for at skifte fra version 1.1 til 1.2.
- Hvis to versioner af en webservice har samme major og minor version men forskellig patch version, skal serviceanvenderen ikke foretage sig noget for at skifte mellem de to versioner.

I denne forbindelse indgår URI'ens faste del for den udstillede REST ressource ikke som en del af det, der versioneres. Serviceudbyderen kan derfor ændre den faste del – f.eks. ændring af serverens DNS navn i en HTTP URI for en REST ressource – uden at dette opfattes som en ikke-kompatibel ændring.

Andre ændringer, som f.eks. ændring af sikkerhedsmodel, ændring af strukturen på URI'en, og ændring af udstillede attributter eller relationer, defineres som en ikke-kompatibel ændring. Andre ikke-kompatible ændringer (der kræver nyt versionsnummer) er f.eks. introduktion af nye obligatoriske parametre eller sletning af parametre.

---

<sup>9</sup> Semantisk versionering er en oversættelse af ”Semantic Versioning”, der er beskrevet her: <http://semver.org>

Navn:	R13. Anvend semantisk versionering af webservices
<b>Retningslinje</b>	Webservices skal anvende semantisk versionering til at tydeliggøre kompatibilitet og overholde versioneringsreglerne.
<b>Rationale</b>	Serviceanvendere skal gøres tydeligt opmærksomme på, hvornår ændringer til en udstillet webservice er kompatibel med eksisterende serviceanvendere, og hvornår dette ikke er tilfældet. Semantisk versionering indeholder flere komponenter i et versionsnummer, der gør dette klart: ændringer til ”major” komponenten i servicens versionsnummer betyder brud på kompatibilitet med eksisterende serviceanvendere.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 og F3. Retningslinjen præciserer OIO serviceprincippet ”Services versioneres” (se [OIOSVC]).
<b>Implikation</b>	<p>Webservices skal garantere kompatibilitet mellem versioner af en webservice ifølge reglerne for semantisk versionering. Webservices kan eventuelt understøtte, at serviceanvenderen kan specificere den ønskede version af webservicen i forbindelse med et kald.</p> <p>En konsekvens af denne versionering er, at følgende ændringer generelt bør medføre en ny major version (listen er ikke udtømmende):</p> <ul style="list-style-type: none"> <li>• En valgfri kaldsparameter bliver obligatorisk</li> <li>• Tilføjelse af en ny obligatorisk parameter ved kald</li> <li>• Tilføjelse af en ny værdi eller nye data i et svar</li> <li>• Fjernelse af værdier i svar</li> <li>• Semantiske ændringer af data i kald eller svar</li> </ul>

Når der sker ikke-kompatible ændringer til udstillede webservices bør både den nye version og den hidtidige version være tilgængelig samtidigt i en overgangsperiode, for at lette overgangen og mindske tidspresset for serviceanvenderne. Konsekvenserne af sådanne ændringer skal generelt minimeres.

Serviceudbyderen skal, jf. retningslinje R12 ovenfor, dokumentere sin strategi for parallel drift.

Navn:	R14. Udstil gamle og nye versioner når en webservice ændres
<b>Retningslinje</b>	Serviceudbyderen skal udstille mindst to forskellige major versioner i en tilstrækkelig overgangsperiode fx mindst 12 måneder, hvis skiftet fra gammel til ny version kræver en væsentlig ændring hos serviceanvenderen.
<b>Rationale</b>	Eksisterende serviceanvendere vil kun undtagelsesvist umiddelbart kunne tilpasse sig til brug af en ændret webservice. Serviceudbyderen må derfor antage at serviceanvenderne har

	behov for en overgangsperiode, hvor både gammel og ny udgave af webservicen udstilles samtidigt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2 og F3.
<b>Implikation</b>	Hvis ændringen udgør mere end blot en anden placering af webservicen (URI'en), bør serviceudbyderen udstille både ny og gammel version af webservicen i en overgangsperiode. Serviceudbyderen skal dokumentere dette, jf. retningslinje R12.

### 3.5 Servicelogning

I forbindelse med integration mellem it-løsninger er det nødvendigt på forlangende at kunne redegøre for kald og svar herpå. Nødvendigheden opstår bl.a. i forbindelse med revisionsformål (hvad har en givet handling fra en bruger forårsaget af adgang til data?). Det opstår også typisk i forbindelse med fejlsituationer (er et givet kald gennemført og med hvilket svar?). Traditionelt er det serviceudbyderen, der foretager logning af modtagne kald samt de svar, der returneres til serviceanvenderen. Da svar imidlertid kan gå tabt på vej tilbage til serviceanvenderen, er det nødvendigt, at både serviceanvender og serviceudbyder foretager behørig logning.

Navn:	R15. Log alle væsentlige kald til webservices
<b>Retningslinje</b>	Alle væsentlige kald til og svar fra webservices skal logges af serviceudbyderen.
<b>Rationale</b>	Det skal sikres, at alle væsentlige kald til en webservice logges. Væsentlige kald omfatter bl.a. kald, der indeholder eller kan indeholde følsomme eller fortrolige data i kaldet eller svaret på kaldet. Kaldet og svaret skal kunne spores for at bl.a. opfylde krav til logning i forbindelse med behandling af persondata er opfyldt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2, F12, og F11.
<b>Implikation</b>	Serviceudbyder skal logge alle kald til en givet webservice, hvor følsomme eller fortrolige informationer behandles i henhold til relevant lovgivning. Serviceudbyder skal desuden overveje om data af ikke-følsom eller ikke-fortrolig natur desuden skal logges. Serviceudbyderen skal dokumentere hvilke kald en givet webservice logger. Webservicen skal desuden foretage logning af alle eksterne kald, der foretages som konsekvens af det modtagne kald, hvor fortrolige eller følsomme data er involverede.



For at sikre at logningen hos serviceanvender og serviceudbyder kan sammenholdes, skal det enkelte kald være unikt identificeret. En sådan identificering har form af en unik transaktionsidentifikator. Serviceanvenderen er ansvarlig for at vælge identifikatorens værdi, da den så kan anvendes til at sikre sammenhæng ved kald til flere webservices. Serviceudbyderens svar skal indeholde identifikatoren, så kald og svar kan sammenholdes.

Navn:	R16. Anvend transaktionsidentifikatorer ved kald og svar
<b>Retningslinje</b>	Webservices skal kræve transaktionsidentifikatorer ved kald, og skal returnere dem ved svar.
<b>Rationale</b>	Det skal være muligt for serviceanvendere og serviceudbydere at spore de individuelle kald til en udstillet webservice samt svar herfra. Ved at inkludere en transaktionsidentifikator i kald og svar, vil dette være muligt for både serviceanvender og serviceudbyder.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1, F12, og F11.
<b>Implikation</b>	Serviceudbyderen skal kræve, at serviceanvenderen medsender en unik transaktionsidentifikator for hvert kald til en udstillet webservice. Transaktionsidentifikatoren kan f.eks. være et universelt unikt id, jf. internet RFC 4122. Webservicens svar skal inkludere transaktionsidentifikatoren fra kaldet. Hvis webservicen udsender forretningshændelser som konsekvens af kaldet fra serviceanvenderen, skal forretningshændelserne inkludere transaktionsidentifikatoren. Serviceanvenderen kan på denne måde detektere konsekvenserne af ”sit eget kald” og eventuelt ignorere forretningshændelsen.

### 3.6 Servicemonitorering

Serviceanvendere har behov for at kunne afgøre om en udstillet webservice er tilgængelig på et givet tidspunkt. Når fejl opstår hos serviceanvenderen i forbindelse med anvendelse af en udstillet webservice, kan adgang til monitorering af anvendte webservices bistå væsentligt ved løsning eller omgåelse af fejl. Serviceudbyderen kan derfor udstille informationer om webservicens tilgængelige til serviceanvendere.

Monitoreringen kan både være inkluderet som en dedikeret (del)funktionalitet af en udstillet webservice eller være udstillet som en sideordnet webservice.



Alternativt kan monitorering understøttes ved at dokumentere en særlig måde at anvende en udstillet webservice på. Dette kunne f.eks. være et fremsøgningskald, der aldrig returnerer data, men svarer med 0 fundne forretningsobjekter, og hvor brugen heraf til monitorering er dokumenteret af serviceudbyderen.

Det centrale i forbindelse med monitorering er, at monitoreringen ikke må foretage opdateringer i data. Logning hos serviceudbyderen er i denne forbindelse ikke at betragte som en opdatering af data.

Navn:	R17. Understøt monitorering af udstillede webservices
<b>Retningslinje</b>	Udstillede webservices skal understøtte, at serviceanvenderen kan monitorere den enkelte webservices tilgængelighed.
<b>Rationale</b>	Serviceanvendere har ved unormale driftssituationer behov for at afgøre om anvendte webservices er tilgængelige eller ikke, og om eventuelle driftsproblemer er placeret hos serviceanvenderen eller hos serviceudbyderen. Udstilling af dedikeret monitoreringsfunktionalitet har typisk større robusthed end de webservices, der anvendes til forretningsprocesserne.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2.
<b>Implikation</b>	Udstillede webservices bør dokumentere, hvordan en serviceanvender kan kontrollere om en webservice er tilgængelig eller ikke. Kontrollen må ikke foretage opdateringer (dvs. må ikke ændre data hos serviceudbyderen), men kan f.eks. realiseres som en simpel søgning. Udstillede webservices kan eventuelt udstille specifik funktionalitet hertil, hvor det garanteres, at webservicen udstiller om både den samt afhængige ressourcer er tilgængelige. Sådant funktionalitet – typisk benævnt en ”ping” metode – har typisk ingen parametre og kræver ingen sikkerhedsmæssige rettigheder hos serviceanvenderen. Ping metoder svarer traditionelt med meget korte svar, typisk blot en statuskode, så serviceanvenderen blot skal forholde sig til, om en webservice er tilgængelig eller ikke.

### 3.7 Servicefejlmeldelser

Når fejl opstår i forbindelse med behandling af et kald til en udstillet webservice, er det vigtigt for efterfølgende analyse af fejlen, at tilstrækkelige informationer om fejlen er leveret til serviceanvenderen. Serviceanvenderen skal selv kunne analysere fejlens detaljer og ved hjælp af den modtagne dokumentation afgøre, hvordan fejlen efterfølgende skal håndteres. Dette stiller væsentlige krav til de informationer, der returneres af en udstillet webservice i forbindelse med fejl.

For det første er det vigtigt, at alle fejl forsynes med en dokumenteret fejlkode (en entydig værdi, f.eks. et tal) og et fejlnavn (en overskrift), så serviceanvenderen kan finde fejlen i servicens dokumentation.

For det andet er det vigtigt, at alle fejlmeddelelser indeholder identifikation af, hvor fejlen er opstået, hvilket både kan være hos serviceudbyderen eller hos et system, der kaldes af serviceudbyderen.

Alle konkret returnerede fejlmeddelelser (dvs. indholdt i et svar fra en webservice) bør ligeledes indikere overfor serviceanvenderen, om kaldet vil kunne gennemføres som tidligere afsendt, hvis det gentages. Det er dog ikke i alle situationer, serviceudbyderen er i stand til at afgøre, om det er tilfældet. Et eksempel, hvor et gentaget kald ikke vil ændre udfaldet, er et kald, der er forkert struktureret eller indeholder ulovlige værdier. Her er det serviceanvenderen, der har begået en fejl, og gentagelse vil resultere i samme fejl som tidligere.

Serviceudbyderen bør dog kun angive, at et gentaget kald ikke giver mening, når serviceudbyderen med sikkerhed ved at gentagelse vil resultere i samme fejl.

Endelig er det vigtigt, at fejlmeddelelsen indeholder så mange detaljer som muligt, dog under hensyntagen til at undgå store fejlmeddelelser grundet båndbreddeforbrug, samt sikkerhedskrav og krav om informationers følsomhed og fortrolighed.

Webservices skal derfor anvende den fællesoffentlige struktur for fejlmeddelelser for webservices. Strukturen publiceres og vedligeholdes på Digitaliseringsstyrelsens hjemmeside, og udgør et bilag til retningslinjerne: Bilag 2 ”Fejlstruktur og fejlkoder for REST Webservices”

Navn:	R18. Returnér servicespecifikke fejl som standardiserede fejlmeddelelser
<b>Retningslinje</b>	Webservices skal returnere servicespecifikke fejl som standardiserede fejlmeddelelser til serviceanvenderen, med dokumenterede fejlkoder, og i overensstemmelse med den fællesoffentlige struktur for fejlmeddelelser.
<b>Rationale</b>	Serviceanvendere er afhængige af præcise og især fyldestgørende fejlmeddelelser, således at serviceanvenderen gøres i stand til at håndtere fejlen enten manuelt eller automatisk. En utilstrækkelig fejlmeddelelse medfører, at serviceanvenderen ikke vil kunne udføre fejlhåndtering. Data eller opdateringer af data kan dermed gå tabt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F2, F6, og F12. Retningslinjen præciserer OIO serviceprincippet ”Services returnerer sigende fejlmeddelelser” (se [OIOSVC]).

<b>Implikation</b>	<p>Fejlmeddelelser vedrørende servicespecifikke fejl returneres i den fællesoffentlige struktur for fejlmeddelelser. Denne indeholder bl.a. følgende informationer:</p> <ul style="list-style-type: none"><li>• En specifik, dokumenteret fejlkode (en entydig værdi, f.eks. et tal) og et fejlnavn</li><li>• Kilden til fejlen (f.eks. ved server URI)</li><li>• Kaldets transaktionsidentifikator, jf. retningslinje R16</li><li>• Angivelse af om fejlen er permanent eller om den er transient (dvs. om serviceanvender ikke bør genforsøge kaldet senere)</li><li>• En kort fejltekst</li><li>• En eller flere konkrete fejloplysninger, f.eks. en liste af fejl fra andre systemer eller kilder, samt hvor hver af disse fejl er opståede.</li></ul> <p>Fejlmeddelelser skal anvendes i kontekst af den valgte transportprotokol, der også returnerer fejlkoder, se retningslinje R33. Det betyder, at transportprotokollens fejlkoder skal bruges korrekt, når servicespecifikke fejl returneres i svar.</p>
--------------------	---

## 4. Retningslinjer for REST webservices

Retningslinjer for REST webservices er opdelt i følgende områder:

- Modellering af REST webservices
- Søgning i REST webservices
- Datarepræsentation i REST webservices
- REST webservice specifikation
- REST webservice versionering
- REST kommunikationsprotokol

Retningslinjerne beskrives efter samme mønster som anvendt i det foregående afsnit 1.

### 4.1 Modellering af REST webservices

REST er, jf. [FIELDING], en arkitekturstil, der anvendes til udstilling af ressourcer i webservices. REST er baseret på de principper og den arkitektur, der også ligger til grund for internettet, med brug og genbrug af få men udbredte standarder og protokoller.

En REST webservice udstiller en *resource*. En ressource er i princippet en betegnelse for enhver form for information. I praksis er en ressource typisk noget, der kan lagres på en computer som f.eks. et dokument, en række i en database, et beregningsresultat, en mediefil mv. I kontekst af fællesoffentlig interoperabilitet er ressourcen typisk data. REST specificerer, jf. [FIELDING], at visse handlinger – oprettelse, læsning, opdatering, sletning mm. kan udføres ”på” ressourcen.

Hvor meget af en ressource, der udstilles i en webservice, skal afvejes overfor retningslinje R01 og R02 om at minimere konsekvenserne af ændringer.

De informationer, der udstilles om ressourcen, udgør en fast aftale (eller en ”kontrakt”) mellem serviceanvender og serviceudbyder. REST arkitekturstilen opererer ikke med kontrakter tilsvarende SOAP protokollen. I praksis er serviceanvender dog afhængig af en række informationer for let at kunne anvende en webservice. Disse informationer opfattes i nærværende sammenhæng som ”kontrakten”, uanset at der ikke foreligger en formel specifikation som i forbindelse med SOAP webservices. Informationerne kan inkludere krav til sikkerhedsmodel, datarepræsentation, m.m.

Navn:	R19. Udstil webservices som REST ressourcer
<b>Retningslinje</b>	REST webservices skal udstille data som ressourcer, der kan læses og evt. oprettes, opdateres og slettes, jf. REST arkitekturstilen.

<b>Rationale</b>	REST arkitekturstilen fastlægger en model for webservices baseret på udstilling af ressourcer med en givet datarepræsentation. Følges denne designmodel ikke, er den pågældende webservice ikke en REST webservice, men en anden type.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6. Retningslinjen præciserer serviceprincipperne ”Services kan anvendes uafhængigt af platform” og ”Services kan anvendes uafhængigt af lokation”, jf. [OIOSVC].
<b>Implikation</b>	I stedet for at udstille forskellige forretningsoperationer på en webservice, er det selve forretningsobjekterne, der udstilles som en web ressource. Ressourcerne identificeres ved en URI. De URI'er, der eksplicit udstilles til omverdenen, vil serviceanvendere forlade sig på. URI'er for ressourcer vil blive opfattet af serviceanvendere som en aftale mellem serviceudbydere og serviceanvender. URI'er for eksplicit udstillede ressourcer kan derfor kun ændres i meget begrænset omfang. Operationer på ressourcen er givet ved valget af kommunikationsprotokol til REST, se retningslinje R33. For HTTPs vedkommende er dette HTTP bl.a. operationerne GET, PUT, POST og DELETE.

I nogle situationer er det ikke data eller entiteter, der udstilles til en serviceanvender, men en forretningsproces eller igangsætning af en forretningsproces. Et eksempel herpå er udsendelse af et brev til en borger. I sådanne situationer bør serviceudbyderen tilstræbe at følge REST arkitekturstilen og modellere og udstille en ressource, der f.eks. repræsenterer de data, der igangsætter forretningsprocessen. I det konkrete eksempel med udsendelse af breve kunne serviceudbyder udstille en service, der opretter en ny forsendelse.

<b>Navn:</b>	<b>R20. Udstil forretningsprocesser som REST ressourcer</b>
<b>Retningslinje</b>	REST webservices skal udstille forretningsprocesser som ressourcer, der kan læses og evt. oprettes, opdateres og slettes i overensstemmelse med REST arkitekturstilen.
<b>Rationale</b>	REST arkitekturstilen fastlægger en model for webservices baseret på udstilling af ressourcer med en givet datarepræsentation. REST arkitekturstilen kan følges i denne situation ved at modellere de data, der igangsætter forretningsprocessen, og udstille dem som en ressource.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6. Retningslinjen præciserer serviceprincipperne ”Services kan anvendes uafhængigt af platform” og ”Services kan anvendes uafhængigt af lokation” (se [OIOSVC]).

<b>Implikation</b>	<p>I stedet for at udstille forskellige forretningsprocesser på en webservice som operationer eller metoder, udstilles en ressource, hvor de data, der igangsætter forretningsprocessen, kan oprettes.</p> <p>Ressourcerne identificeres ved en URI.</p> <p>Operationer på ressourcen er givet ved valget af kommunikationsprotokol til REST, se retningslinje R33. Konkret betyder det f.eks. for HTTPs vedkommende, at HTTP operationerne PUT, GET og DELETE f.eks. kan anvendes til at igangsætte en forretningsproces, hente status, eller standse forretningsprocessen.</p>
--------------------	--

Udstillede REST ressourcer bør struktureres ud fra den bagvedliggende forretningsmodellering. De begreber og informationsentiteter, forretningsmodelleringen af et givet domæne har afdækket, bør anses som kandidater til udstillede ressourcer. Dette uddybes i bilag 5 ”Udstilling af forretningsobjekter som REST ressourcer.” Forretningsmodelleringen kan være udført efter de fællesoffentlige regler for modellering, jf. [MODEL]. Hvor dette ikke er tilfældet, kan udgangspunktet være en model for domænet.

Navn:	R21. Modellér REST ressourcer med udgangspunkt i forretningsmodellering
<b>Retningslinje</b>	REST ressourcer skal struktureres med udgangspunkt i den forretningsmodellering, der er udført efter de fællesoffentlige regler for begrebs- og datamodelisering [MODEL].
<b>Rationale</b>	Modellering af ressourcer efter [MODEL] sikrer ensartet sprogbrug og fælles grundlag for at fastlægge granularitet og serviceboundaries. Det letter og tydeliggør den menneskelige forståelse af funktionalitet, syntaks og semantik for en webservice, hvis webservicens design har udgangspunkt i en modellering af forretningen. Samme forståelse af informationen sikres hos serviceanvender og serviceudbyder, og dermed etableres semantisk interoperabilitet.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6. Retningslinjen omfatter og udvider OIO serviceprincippet ”Services understøtter forretningen” (se [OIOSVC]).
<b>Implikation</b>	<p>Udstillede ressourcer skal have oprindelse i en modellering af forretningens begreber og informationer.</p> <p>Udgangspunktet for ressourcerne skal være forretningsmodelleringens ”forretningsobjekter” med egenskaber. Relaterede entiteter udstilles i udgangspunktet som separate entiteter, se også retningslinje R25.</p> <p>Forretningsmodellering kan være en konsekvens af domænerelateret standardisering, f.eks. HL7/FHIR på sundhedsområdet.</p>

	Hvis en samlet forretningsmodellering ikke er gennemført, er et alternativt udgangspunkt de vigtigste entiteter fra en anden modellering. Der henvises til retningslinje R29 vedrørende repræsentation af ressourcers data.
--	--

De ressourcer, der udstilles, bør følge navnene fra forretningsmodellen, så koblingen mellem forretningsmæssig og teknisk forståelse i forbindelse med webservice tydeliggøres.

Navn:	R22. Navngiv REST ressourcer ud fra forretningsmodellens begreber
<b>Retningslinje</b>	Navngivning af REST ressourcer skal tage udgangspunkt i den forretningsmodellering, der er udført efter de fællesoffentlige regler for begrebs- og datamodellering [MODEL].
<b>Rationale</b>	Et udgangspunkt i forretningsmodellens begreber sikrer ensartet navngivning og fælles grundlag for genbrug og fremsøgning. Forretningsmodelleringen afdækker forretningsobjekter og forklarer relationen mellem forskellige forretningsobjekter. Hvis ressourcerne, der kan tilgås via en REST webservice, anvender sigende navne, har koblingen til forretningsforståelsen et fast fundament.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1 og F6.
<b>Implikation</b>	Ressourcer navngives fra domænets begreber, f.eks. ”sager”, ”journalnotater”, ”parter”. Begreberne kan være en konsekvens af domænerelateret standardisering, f.eks. HL7/FHIR på sundhedsområdet. Se også retningslinje R23 vedrørende konkret navngivning.

En ressource adresseres og identificeres med en ressourceidentifikator i form af en URI, der navngiver ressourcen og angiver hvor ressourcen kan findes.

Ifølge REST arkitekturstilen angives ressourcen som et navneord, f.eks. ”/sager” eller ”/personer”, og et ”scheme” (typisk protokol og placeringen af ressourcen) angives foran ressourcens navn, f.eks. ”https://it.system.dk/sager”.

Konkrete instanser af ressourcer kan yderligere specificeres som f.eks.

”https://it.system.dk/sager/123” eller ”https://it.system.dk/per-

soner/345". Særlig opmærksomhed henledes på anvendelse af danske specialtegn i navngivningen af ressourcen. Hvis disse anvendes vil det kræve en særlig kodning af URI'en<sup>10</sup>, hvilket gør denne sværere at læse.

Navn:	R23. Navngiv REST ressourcer ud fra REST arkitekturstilen
<b>Retningslinje</b>	REST ressourcer skal navngives som navneord i flertal jf. REST arkitekturstilen, og skal navngives så der ikke kræves (URI) kodning af navnet.
<b>Rationale</b>	For at følge REST arkitekturstilen skal ressourcer navngives med navneord, i stedet for at navngive efter operationsnavne. Ved ikke at kræve en kodning af URI'en er det let for anvendere at læse og lagre ressourcers URI'er menneskelæsevenligt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6, F1 og F8.
<b>Implikation</b>	Ifølge REST arkitekturstilen vil ressourcer typisk udstilles som en URI, der indeholder ressourcens navn i flertal. En sags ressource kunne således have en URI, der ligner <code>"/http://server.dk/sager"</code> (bemærk understregning af flertalsform). Da de danske specialtegn "æ", "ø" og "å" vil kræve URI kodning, bør ressourcer navngives uden disse, for at URI'erne er lette at læse for mennesker.

REST ressourcers identifikatorer er et væsentligt element i den "kontrakt", en REST webservice udstiller. Hvis identifikatoren på en ressource ændres, vil serviceanvenderen ikke længere kunne finde den pågældende ressource. Det er derfor vigtigt, at en ressourcers identifikator ikke ændrer sig i løbet af ressourcens levetid. Hvis en ændring foretages, vil det kræve en kompenserende ændring hos serviceanvenderen.

Det er vigtigt, at ressourcers identifikatorer ikke indeholder følsomme eller fortrolige informationer, eller sikkerhedsrelaterede informationer. Identifikatorerne opbevares og logges typisk i forbindelse med serviceanvendelse. Det er derfor u hensigtsmæssigt, at identifikatorerne indeholder sådanne informationer, idet logningen i så fald bliver følsom eller fortrolig.

Navn:	R24. Udstillede REST ressourcer har unikke, sikre identifikatorer
<b>Retningslinje</b>	Alle udstillede REST ressourcer skal have unikke identifikatorer (URI'er), der er stabile og sikre.

<sup>10</sup> URI kodning er specificeret i internet RFC 3986, og anvender en % (procent) notation for ikke tilladte karakterer, som f.eks. mellemrum, der bliver til "%20".



<b>Rationale</b>	<p>Unikke identifikatorer sikrer at data kan genfindes i ressourcens levetid.</p> <p>REST ressourcers unikke identifikatorer skal være unikke., hvorved forstås, at de kun må udpege én ressource.</p> <p>Identifikatorerne skal være stabile, hvorved forstås, at de dermed ikke må ændre sig i ressourcens levetid. Det skal være muligt for serviceanvendere at opbevare udstillede ressourcers identifikatorer.</p> <p>Identifikatorerne skal være sikre, hvorved forstås, at de ikke må indeholde personfølsomme data, da identifikatorerne typisk vil indgå i den krævede logning af adgang til ressourcen (se også retningslinje R35).</p>
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F1.
<b>Implikation</b>	Serviceudbyderen er selv ansvarlig for at definere identifikatoren for en givet ressource. Identifikatoren kan være et UUID, men dette er ikke et krav (et teknisk løbenummer kan anvendes i stedet, blot det er unikt). Personnumre kan ikke anvendes som unikke identifikatorer, da en persons personnummer kan ændre sig i personens levetid. Logning af en person (resources) URI der indeholder et personnummer medfører, at logningen bliver personfølsom, og dermed underlægges regler om sletning, der vanskeligt kan håndhæves.

Forretningsobjekter, der udstilles som REST ressourcer, vil naturligt have relationer til andre forretningsobjekter. Disse relaterede entiteter bør udstilles som separate (under)ressourcer, som kan tilgås direkte ved brug af en passende identifikator (URI). Serviceanvendere gives dermed mulighed for at tilføje, ændre eller fjerne relaterede entiteter uden at skulle opdatere hele ressourcen.

Navn:	R25. Udstil REST ressourcers relaterede entiteter
<b>Retningslinje</b>	REST ressourcer skal udstilles med relationer til de forretningsobjekter, der er relateret til eller indeholdt i forretningsobjektet der udstilles jf. den forretningsmodellering, der er udført efter de fællesoffentlige regler for begrebs- og datamodellering [MODEL].
<b>Rationale</b>	REST ressourcer repræsenterer begreber i forretningsmodelleringen. Forretningsmodelleringen afdækker relationer mellem begreber. Disse relationer bør kunne tilgås som underressourcer, jf. REST arkitekturstilen, og gør det forståelsesmæssigt lettere og teknisk mere effektivt at anvende web-servicen.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F1.

<b>Implikation</b>	REST webservices bør udstille relaterede forretningsobjekter som underressourcer, f.eks. <code>"/sager/&lt;sagsid&gt;/journalnotater/&lt;notatid&gt;"</code> . Navngivningen af underressourcen bør jf. retningslinje R22 følge navne fra modelleringen.
--------------------	--

REST arkitekturstilen fastlægger at hypertext (dvs. indhold med aktive links, der kan navigeres efter) er måden hvorpå ressourcer tilgås. REST webservices bør derfor informere om sig selv og om relevante andre ressourcer via hypertext links.

REST webservices udstiller eksplicit en række ressourcer med velkendte URI'er. I det omfang, relaterede objekter udstilles via hypertext links, er det en fordel for serviceudbyderen ikke at forpligte sig til disse hypertext links' værdier. Hvis serviceanvenderen ikke må opbevare URI'erne til relaterede objekter men skal fremfinde dem hver gang, får serviceudbyderen bedre muligheder for at foretage ændringer uden at serviceanvenderen påvirkes.

Navn:	R26. Returnér REST ressourcer med hypertext links
<b>Retningslinje</b>	REST ressourcer skal returnere hypertext links til alle relevante informationer om ressourcen, når ressourcen tilgås, jf. REST arkitekturstilen.
<b>Rationale</b>	REST arkitekturstilen (HATEOAS) samt [MATURITY] påpeger, at REST webservices skal returnere metainformationer om ressourcer via hypertext links. Hvis ressourcer returnerer relevante links som metadata, behøver serviceanvendere ikke at kende de konkrete links på forhånd. Dette giver serviceudviklere frie hænder til at ændre URI'er på de forretningsobjekter, der ikke direkte er udstillet til serviceanvendere.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F1.
<b>Implikation</b>	REST ressourcer bør returnere deres egen URI, links til en specifikation af deres repræsentation, links til formidling af URI'er til relaterede objekter, og links til de operationer ressourcen aktuelt understøtter. Disse links bør returneres ved for hvert link at angive linket med dets relation <sup>11</sup> til ressourcen.

---

<sup>11</sup> En fællesoffentlig standard kan evt. fastlægges for at sikre ensartethed.

## 4.2 Søgninger i REST webservices

REST webservices vil ofte udstille ressourcer, der er samlinger af forretningsobjekter, f.eks. forvaltningssager, aktører, eller dokumenter. Serviceanvenderne har typisk ikke brug for altid at hente alle de konkrete instanser, og har derfor behov for at udtrykke søgning, hvis ikke URI'en for den konkrete instans er kendt på forhånd.

En REST webservice, der understøtter søgning, bør udstille at resources URI kan suppleres med ”parametre” til søgning eller sortering, der anvendes ved læsning af ressourcer (ved brug af HTTP GET).

Søgeparametrene kan understøtte, at der angives værdier for ønskede attributter eller tilstedeværelsen af relationer. Når de udvalgte konkrete instanser er fremsøgt, kan serviceanvenderen have behov for at angive en sortering, eller en præcisering af hvilke informationer serviceanvenderen har brug for. Standardiserede søgeparametre på tværs af fællesoffentlige webservices vil gøre det forståelsesmæssigt lettere at anvende sådanne webservices.

Understøttelsen af søgning skal dog afvejes overfor retningslinje R01 og R02 om at minimere funktionalitet og målrette webservices.

Navn:	R27. Anvend standardiserede REST fremsøgningsparametre
<b>Retningslinje</b>	REST webservices skal understøtte standardiserede fremsøgningsparametre i det omfang, søgning understøttes af webservicen.
<b>Rationale</b>	For at gøre det lettere at anvende en webservice for serviceanvendere, bør fremsøgninger i samlinger af forretningsobjekter anvende standardiserede søgeparametre. Ensretning af parametrene betyder at serviceanvendere kan genbruge eksisterende erfaringer med søgning.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F2.
<b>Implikation</b>	<p>Standardiseringen bør inkludere parametre for udvælgelse (query), sortering af resultater, selektion af returnerede attributter for fremsøgte ressourcer, samt om relaterede ressourcer skal returneres.</p> <ul style="list-style-type: none"> <li>• Søgning i ressourcer angives i URI som en query string med ”søgeoperator” ’q’ (dvs. ”?q=alder%3d20”)</li> <li>• Sortering ved resourcesøgninger angives i URI som query string med ”søgeoperator” ’sort’ (dvs. ”?sort=alder”)</li> </ul>

	<ul style="list-style-type: none"> <li>• Selektion af attributter ved ressourcøsøgninger angives i URI som query string med ”søgeoperator” ’fields’ (dvs. ”?fields=navn,adresse”)</li> <li>• Embedding af relaterede objekter ved ressourcøsøgninger angives i URI som query string med ”søgeoperator” ’embed’ (dvs. ”?embed=barn”)</li> </ul> <p>De faktisk understøttede søgemuligheder skal målrettes de faktiske behov, jf. retningslinjerne R01 og R02.</p>
--	--

Når serviceanvendere bruger REST webservices, der udstiller samlinger med mange objekter, vil serviceanvenderen typisk ønske at kunne fremsøge objekterne gradvist for at forbedre svartider på søgning. En REST webservice kan understøtte dette ved at tillade ”paginering” (søgeresultater kan indlæses ”sidevist”). Paginering er en velkendt teknik, hvor webservicen giver serviceanvenderen mulighed for at indikere et antal ønskede objekter, samt hvorfra søgningen skal ”fortsætte”.

Navn:	R28. Understøt søgning med delresultater
<b>Retningslinje</b>	REST webservices, der udstiller store samlinger af forretningsobjekter, skal understøtte paginering i forbindelse med den søgning, webservicen understøtter.
<b>Rationale</b>	En webservice (f.eks. en REST ressource), der udstilles som en samling af objekter, og som kan indeholde et stort antal forretningsobjekter eller relaterede forretningsobjekter, bør understøtte delvis afhentning af søgeresultater (dvs. paginering), så serviceanvender ikke altid skal hente hele det fremsøgte.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F2.
<b>Implikation</b>	<p>Webservices, der tilgår store mængder af objekter, bør understøtte pagineringsparametre under fremsøgning. Pagineringen bør anvende en eksplicit ”cursor” (f.eks. element identifikator eller position), der skal medsendes fra serviceanvenderen. Serviceudbyderen kan lade sig inspirere af internet RFC 5005<sup>12</sup>. Det samlede antal objekter der udgør søgningens resultat bør medsendes, hvis dette kendes af webservicen.</p> <p>Serviceudbyderen bør dokumentere hvordan pagineringen påvirkes hvis der sker ændringer til samlingen af objekter mens pagineringen pågår.</p> <p>De understøttede muligheder for paginering skal målrettes de faktiske behov, jf. retningslinjerne R01 og R02.</p>

<sup>12</sup> Se f.eks.: <https://tools.ietf.org/html/rfc5005>

### 4.3 Datarepræsentation i REST webservices

REST webservices kan i princippet udstille ressourcers data med en vilkårlig datarepræsentation. I praksis repræsenteres ressourcer ofte i JSON format eller som et XML dokument.

JSON vurderes både generelt velegnet, og i sammenligning med XML særlig velegnet, når behovet er kommunikation med en webbrowser. JSON har yderligere den fordel, at det er relativt kompakt.

XML er har stor udbredelse i fællesoffentlige webservices (der konkret udstilles som SOAP services). Mange udviklingsteknologier understøtter validering af XML (specificeret via XSD), og udstilling som domæneobjekter i programmeringssproget (via autogenerated kode).

Det vurderes dog vigtigst, at den repræsentation en ressource udstilles med er relevant for forretningsdomænet.

Nogle domæner som f.eks. geodataområdet og sundhedsområdet har i væsentligt omfang egne standarder (henholdsvis OGC og HL7/FHIR). En domænespecifik, standardiseret repræsentation foretrækkes derfor, da de syntaktiske og semantiske forhold ofte er afklarede og veldokumenterede.

Hvis der på et forretningsområde ikke findes en standard, er JSON eller XML velegnet som generel datarepræsentation, da mulighederne for repræsentation af komplekse datastrukturer normalt er tilstrækkelige i disse formater.

Navn:	R29. Repræsenter REST ressourcer i et standardiseret dataformat
<b>Retningslinje</b>	REST ressourcers data skal repræsenteres i et standardiseret dataformat for forretningsdomænet. Hvis et sådant dataformat ikke findes, skal en repræsentation i JSON eller XML anvendes.
<b>Rationale</b>	Serviceudbyderen skal udstille forretningsobjekter til serviceanvendere. Hvis der til forretningsobjekterne findes standarder for dataformater inden for serviceudbyderens domæne – f.eks. HL7/FHIR for sundhedsdomænet – bør disse anvendes for at sikre, at serviceanvender har et veldefineret grundlag for at forstå data. Hvis data oversættes er der en risiko for at miste informationer, eller at serviceanvenderen ikke forstår og behandler data korrekt.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6, F8, F9, og F2.

	Retningslinjen er en forlængelse OIO serviceprincippet ”Services baseret på standarder” (se [OIOSVC]).
<b>Implikation</b>	Hvis et domænerelateret dataformat er anvendeligt bør webservices anvende dette. Ved standardiseret dataformat forstås i denne forbindelse også dataformater, der har bred anvendelse men ikke formelt er standardiserede. Der bør i denne forbindelse tages hensyn til om et givet standardiseret dataformat er velegnet til transmission via REST og HTTP.

REST webservices skal udstille og deklarerer hvilken datarepræsentation de tilgængelige ressourcer har. REST webservices, der tilgås via http, har yderligere den mulighed at alternative datarepræsentationer kan tilbydes. En ressource kan således deklarerer, at den udstiller både JSON og XML. En serviceanvender kan ved adgang til ressourcen via HTTP indikere hvilke repræsentationer, der foretrækkes (med prioritering).

Navn:	R30. Deklarér REST ressourcer datarepræsentationer
<b>Retningslinje</b>	REST ressourcer skal i deres kommunikation deklarerer hvilke datarepræsentationer (f.eks. JSON, XML), de understøtter.
<b>Rationale</b>	Hvis en REST ressource deklarerer de understøttede repræsentationer, er det muligt for serviceanvenderen at vælge den bedst egnede, således at webservicen målretter sig serviceanvenderens behov.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6, F9, og F2.
<b>Implikation</b>	REST ressourcer bør via en HTTP header altid tilkendegive, hvilke repræsentationer ressourcen understøtter, når ressourcen returneres fra serviceudbyderen, f.eks. ved et opslag eller en opdatering.

Når REST webservices udstiller tekst – navne, beskrivelser, mv. – er det vigtigt at tekstens individuelle tegn overføres korrekt, især i forhold til de danske speci-tegn ’æøå’, men også tegn der ofte anvendes i navne, f.eks. ’äëöüé’. Tekstkodning er ofte en udfordring med især ældre it-løsninger, der udstiller webservices, eller når data fra en database udstilles direkte (da databaserne sjældent anvender en internationaliseret tekst kodning internt).

Navn:	R31. Overfør tekst i en internationaliseret repræsentation
<b>Retningslinje</b>	Tekst, herunder navne, beskrivelser mm. skal overføres mellem serviceanvender og serviceudbyder i et internationaliseret overførselsformat, UTF-8 eller tilsvarende, med mindre en anvendt domænespecifik standard tilsiger andet.

<b>Rationale</b>	De danske specialtegn samt tegn fra EU og andre lande anvendes bredt i webservices. Serviceudbyder og serviceanvender skal vide sig sikre på, at tegn overføres korrekt. UTF-8 standarden skal anvendes hertil, da den sikrer korrekt overførsel og er i stand til at repræsentere alle nødvendige tegn.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6 og F9. Retningslinjen præciserer OIO serviceprincippet ”Services baseret på standarder” (se [OIOSVC]).
<b>Implikation</b>	Udstillede webservices skal sikre, at tekst der skal returneres til eller modtages fra serviceanvender konverteres, såfremt teksten ikke efterfølgende opbevares og behandles i UTF-8. Hvis teksten skal konverteres, er webservicen ansvarlig for konverteringen. En undtagelse i denne forbindelse er, hvis en domænespecifik standard, der anvendes til webservicens data repræsentation, fastlægger et andet format.

#### 4.4 REST webservice specifikation

En REST webservice skal udstille specifikationen af de data, der indgår i webservicen. Når en ressource tilgås, kan webservicen returnere en definition af det skema, der definerer ressourcens data, f.eks. via et hypertext link i svaret. Et anvendelsesystem er dermed i stand til selv at hente en maskinlæsbar datarepræsentation.

Den maskinlæsbare datarepræsentation skal udtrykkes via et standardiseret skema for REST services. Dette skema fremgår af den fællesoffentlige dokumentationsramme for webservices. Den maskinelæsbare datarepræsentation er en del af dokumentationen for en webservice.

Navn:	R32. Dokumentér REST webservices via et standardiseret skema
<b>Retningslinje</b>	Webservices skal dokumentere de udstillede data via et standardiseret skema for REST services, jf. den fællesoffentlige dokumentationsramme for webservices.
<b>Rationale</b>	Det skal være klart for serviceanvenderne hvilke informationer, der udstilles af en webservice, og hvilken semantik og syntaks de udstillede data har. Dette bør dokumenteres ved brug af et standardiseret skema, således at serviceudbyder og serviceanvender har samme forståelse.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F6. Retningslinjen præciserer OIO serviceprincippet ”Services er velbeskrevne” (se [OIOSVC]).
<b>Implikation</b>	REST webservices bør anvende standardiserede dataformater jf. retningslinje R29. Hvis der derfor findes en standard bør



	<p>dokumentationen henviser hertil. Hvis der ikke forefindes en standard, bør de udstillede REST ressourcer f.eks. dokumenteres via en JSON specifikation, der både indeholder ressourcers URI, ressourcers data (attributter) samt hvilke relationer de enkelte ressourcer udstiller.</p> <p>Skemadefinitionen af den udstillede REST webservice skal inkludere en reference til den bagvedliggende datamodelering, hvis en sådan findes, jf. retningslinje R21.</p>
--	---

## 4.5 REST kommunikationsprotokol

REST arkitekturstilen kræver ikke HTTP, men HTTP er den mest anvendte kommunikationsprotokol til REST webservices. HTTP er velegnet til REST webservices, og der findes væsentlig værktøjs- og teknologiunderstøttelse af REST via HTTP.

HTTP består af en request struktur og en response struktur. I request strukturen angiver serviceanvenderen HTTP metoden (verbet), der skal gøre noget ved ressourcen, som er udpeget vha. en URI. I request headeren angives desuden parametre og andre metadata for forespørgslen. Eksempelvist kan der vha. authorization vedlægges et sikkerhedstoken. Det kan også være parametre for cache, tekstkodning, dato mv. I request body angives en evt. inputstruktur for kaldet.

I response-strukturen angives en HTTP statuskode. I response headeren angives parametre og metadata for svaret. I response body sendes ressourcens repræsentation, som er det faktiske beskedindhold, typisk i form af en XML eller JSON-struktur. Den returnerede repræsentation af ressourcen bør afgøres vha. *content negotiation*, der er en HTTP mekanisme for at aftale formatet på de overførte data, jf. retningslinje R34 nedenfor.

Den konkrete brug af HTTP verberne til REST webservices skal følge den fællesoffentlige specificering af brug af HTTP til REST, der publiceres og vedligeholdes på Digitaliseringsstyrelsens hjemmeside, og udgør et bilag til retningslinjerne i dette dokument.

<b>Navn:</b>	<b>R33. Anvend HTTP som fællesoffentlig REST kommunikationsprotokol</b>
<b>Retningslinje</b>	REST webservices skal anvende HTTP som kommunikationsprotokol ved kald fra serviceanvender til den udstillede webservice, og følge den fællesoffentlige specificering af brug af HTTP til REST.
<b>Rationale</b>	HTTP protokollen er standardiseret, og generelt anbefalet i forbindelse med udstilling af REST webservices. HTTP pro-



	tokollen har tilstrækkelig funktionalitet til at understøtte behovene for overførsel/transmission af data i forbindelse med kald af webservices.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1, F6 og F8. Retningslinjen præciserer OIO serviceprincippet ”Services baseret på standarder” (se [OIOSVC]).
<b>Implikation</b>	REST webservices anvender HTTP verberne GET, POST, PUT, DELETE, OPTIONS, HEAD og PATCH, når ressourcer tilgås. Serviceudbyderen skal implementere HTTP verberne som atomare handlinger, og kun anvende HTTP statuskoder, som det fremgår af den fællesoffentlige specificering af brug af HTTP til REST. REST webservices anvender HTTP kommunikationsprotokollens headere til at overføre særlige informationer om kaldet eller svaret, herunder transaktionsidentifikatoren jf. retningslinje R16.

HTTP indeholder en række mekanismer, der bidrager til effektiv brug af båndbredde, datarepræsentation, mv. Disse mekanismer skal anvendes af serviceanvender og serviceudbyder for at sikre effektiv kommunikation.

Navn:	R34. Anvend HTTPs mekanismer til effektiv kommunikation
<b>Retningslinje</b>	Serviceudbyderen skal anvende HTTP således, at effektiv kommunikation med REST webservices understøttes.
<b>Rationale</b>	Hvis serviceudbyderen tillader brug af bestemte HTTPs mekanismer kan kommunikationen mellem serviceanvender og serviceudbyder effektiviseres. Understøttelse af forhandling af ønsket datarepræsentation mellem webservicen og serviceanvenderen målretter webservicen til den konkrete brug. Understøttelse af komprimering af forespørgsel og svar reducerer krav til netværksbåndbredden.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1, F6 og F8.
<b>Implikation</b>	HTTPs ”content negotiation” mekanisme anvendes for at aftale data repræsentation mellem serviceanvender og serviceudbyder. HTTP kompression af forespørgsler og svar skal understøttes, eventuelt implementeret i selve HTTP serveren <sup>13</sup> .

<sup>13</sup> Flere gængse HTTP servere kan opsættes til komprimering og dekomprimering af data, f.eks. Apache HTTPd.

REST webservices bør tilskynde en sikker anvendelse af HTTP. Webservices bør kræve sikre forbindelser (f.eks. HTTPS, eller kryptering og autentifikation via en anden mekanisme). De URI'er der anvendes til ressourcer bør ikke indeholde fortrolige eller følsomme data. Logning af HTTP typisk logger den tilgæede URI, og dermed kan fortrolige eller følsomme data persisteres i HTTP infrastrukturens logning.

Endelig bør det sikres, at eventuelle HTTP servere der er placerede ”foran” udstillede webservices ikke logger den dynamiske del af URI'en (query string), da en søgning eventuelt kunne søge på følsomme eller fortrolige data (f.eks. personnummer).

Navn:	R35. Anvend HTTP sikkert til REST ressourcer
<b>Retningslinje</b>	REST webservices skal anvende HTTP sikkert ved udstilling af ressourcer.
<b>Rationale</b>	Serviceudbyderen skal tilskynde, at HTTP anvendes på en sikker måde af serviceanvenderen. Dette kan serviceudbyderen gøre ved bl.a. at kræve sikker (krypteret og autentificeret) HTTP, og ved at sikre, at URI'er ikke i sig selv indeholder følsomme eller fortrolige data.
<b>Understøttede forretningsbehov</b>	Retningslinjen er affødt af forretningsbehov F1 og F10.
<b>Implikation</b>	Webservices bør returnere specifikke fejlkoder, hvis serviceanvenderen bruger usikker HTTP (dvs. uden kryptering og autentificering) i stedet for automatisk at omstille serviceanvenderen til en sikker HTTP. REST webservices bør kun udstille URI'er, der ikke indeholder nøgler af følsom karakter, f.eks. personnumre eller sikkerhedsinformationer. Hvis udstillede REST webservices anvender en ”foranstillet” HTTP server, må denne ikke logge HTTP ”query string” eller andre evt. følsomme eller fortrolige data, da denne logning ellers skal håndteres som følsomme data.

## 5. Referenceliste

ID	Kilde
[DSTRATEGI]	”Den fællesoffentlige digitaliseringsstrategi 2016-2020”, Digitaliseringsstyrelsen. <a href="https://www.digst.dk/Strategier/Strategi-2016-2020">https://www.digst.dk/Strategier/Strategi-2016-2020</a>
[FIELDING]	” <i>Architectural Styles and the Design of Network-based Software Architectures</i> ”, Roy Thomas Fielding, Dissertation 2000
[HVIDBOG]	” <i>Den digitalt sammenhængende offentlige sektor – Hvidbog om arkitektur for digitalisering</i> ”, version 1.0, juni 2017, Digitaliseringsstyrelsen. <a href="http://arkitektur.digst.dk/mandat-og-styring/hvidbog-om-faellesoffentlig-digital-arkitektur">http://arkitektur.digst.dk/mandat-og-styring/hvidbog-om-faellesoffentlig-digital-arkitektur</a>
[MATURITY]	” <i>Richardsons maturity model</i> ”, Leonard Richardson 2008, (webressource) <a href="https://www.crummy.com/writing/speaking/2008-QCon/act3.html">https://www.crummy.com/writing/speaking/2008-QCon/act3.html</a>
[MICROSERVICES]	” <i>Microservices</i> ”, Martin Fowler 2014, (webressource) <a href="https://martinfowler.com/articles/microservices.html">https://martinfowler.com/articles/microservices.html</a>
[MODEL]	” <i>Fællesoffentlige regler for begrebs- og datamodellering</i> ”, version 1.0, juni 2017, <a href="https://arkitektur.digst.dk/metoder/regler-begrebs-og-datamodellering">https://arkitektur.digst.dk/metoder/regler-begrebs-og-datamodellering</a>
[OIOIDWS]	” <i>OIO IDWS REST profile</i> ” version 1.0, Digitaliseringsstyrelsen februar 2016
[OIOMØNSTRE]	” <i>Integrationsmønstre</i> ”, Digitaliseringsstyrelsen, (webressource) <a href="http://arkitekturguiden.digitaliser.dk/integrations-moenstre">http://arkitekturguiden.digitaliser.dk/integrations-moenstre</a>
[OIOREST]	” <i>OIOREST i praksis – guidelines baseret på eksempler</i> ”, It- og Telestyrelsen, marts 2009
[OIOSVC]	” <i>Serviceprincipper</i> ”, Digitaliseringsstyrelsen, (webressource) <a href="http://arkitekturguiden.digitaliser.dk/service-principper">http://arkitekturguiden.digitaliser.dk/service-principper</a>
[REGEL7.1]	Arkitekturregel 7.1: Design og udstil snitflader efter fælles integrationsmønstre og tekniske standarder <a href="http://arkitektur.digst.dk/principper-og-regler/princip-7-it-loesninger-samarbejder-effektivt/arkitektur-regel-71-design-og">http://arkitektur.digst.dk/principper-og-regler/princip-7-it-loesninger-samarbejder-effektivt/arkitektur-regel-71-design-og</a>

**Styregruppen for data og arkitektur**