

Fornyings- og administrasjonsdepartementet
v/ Hans Erik Gravdahl Sørensen
Postboks 8004 Dep.
0030 Oslo

Bekk Consulting AS
ved/ Trond Arve Wasskog
Postboks 134 Sentrum
0102 Oslo

Felles elektronisk tjenesteyting i offentlig sektor - høring av arbeidsgrupperapport om felles IKT-arkitektur

Det vises til Fornyings- og Administrasjonsdepartementets brev av 25.06.08 hvor det etterspørres en høringsuttalelse på rapporten "Felles IKT-arkitektur i offentlig sektor", ref 200701034-/HES. Vi vil med dette gi våre innspill til rapporten.

Generelt: Om gjenbruk

Gjenbruk blir nevnt som målsetning flere steder i dokumentet. Vi mener forventningen til gjenbruk bør reduseres, samt at kostnaden ved å utvikle for gjenbruk må synliggjøres tydelig:

- Å designe for gjenbruk er kostbart og tidkrevende. Undersøkelser viser at design for gjenbruk koster minst 3 ganger så mye som å designe for konkrete behov. Denne kostnaden må synliggjøres og budsjetter må ta høyde for denne kostnaden ved planlegging og tildeling.
- Gjenbruk bør baseres på faktiske behov. Spekulativt gjenbruk må unngås; dette er sannsynligvis blant de mest kostnadsdrivende elementene i utvikling av IKT-prosjekter.
- Gjenbruk er primært mulig og kostnadseffektivt på teknisk nivå. Eksempler på dette er rammeverk som Java og .NET API, Spring Framework og andre generelle rammeverk.
- Gjenbruk på funksjonelt nivå er vanskelig, kostbart, tidkrevende og krever sterk koordinering. Krever samkjøring av behov, domenebegreper, budsjetter og planer på tvers av sektorer og etater. Gjenbruk på dette nivået vil i praksis være umulig.
- Med hensyn til felleskomponenter er felles Autentisering (eID) og autorisering mulig. På den andre siden er felles Skjemamotor sannsynligvis for domenespesifikk til å kunne gjenbrukes effektivt.

Arkitekturprinsipper (jf. kap. 3 i rapporten)

Generelt om arkitekturprinsipper

Vi anbefaler å moderere prinsippene *Fleksibilitet* og *Skalerbarhet*. Det redegjøres for dette lengre ned i dokumentet. I tillegg foreslår vi å legge til prinsippene *Enkelhet*, *Testbarhet* og *Vedlikeholdbarhet*.

Formålet er mer fokus på kostnader og tidsbruk ved utvikling og forvaltning av løsninger.

Nytt prinsipp: Enkelhet

Definisjon:

Enkelhet er det motsatte av kompleksitet og betegner gjerne renhet og klarhet. Definisjonen av enkelhet er egenskapen å være enkel, ublandet og usammensatt. Enkelhet gjør arkitekturen lettere å forklare, forstå, og vedlikeholde.

Prinsipp:

Enkelhet skal være grunnleggende prinsipp for all arkitektur, infrastruktur, løsninger og tjenester. Dette betyr å redusere kompleksitet i utvikling, produksjon og drift av løsninger, tilstrebe reduksjon av antall komponenter, avhengigheter og varianter samt unngå spesialtilpasninger. Unødvendig kompleksitet er en helt sentral og ofte nødvendig kostnadsdriver i IKT-prosjekter.

Nytt prinsipp: Testbarhet

Definisjon:

Testbarhet betyr mulighet til å lage automatiserte tester for alle komponenter. En automatisert test for en komponent skal kunne kjøres selvstendig, være repeterbar og bruke kort tid. For arkitekturløsninger og rammeverk betyr testbarhet at enkeltkomponenter må kunne testes i isolasjon, utenfor opprinnelig kontekst. Testbarhet gjør det mulig å lage et sett av automatiserte tester som et sikkerhetsnett for løsningen, effektiviserer vedlikehold og reduserer kostnader i hele livssyklusen.

Prinsipp:

Alle komponenter, rammeverk og løsninger skal være testbare. Automatiserte tester utvikles samtidig med løsningen, leveres sammen med løsningen og forvaltes på lik linje med løsningen i livssyklusen til løsningen. Testbarhet i henhold til moderne testprinsipper og -metodikk er allerede et svært utbredt fenomen innen systemutvikling, og helt avgjørende for sikre kvalitet og minimere kostnader innen IKT-prosjekter.

Nytt prinsipp: Vedlikeholdbarhet

Definisjon:

Formålet med vedlikeholdbarhet er å redusere kostnad og tidsbruk ved utvikling, vedlikehold og forvaltning. Dette krever høy teknisk kvalitet i løsningen på områder som navngiving, komponent- og funksjonsdesign, kodestandard, koherens og avhengigheter. Vedlikeholdbarhet henger også tett sammen med prinsippet om Enkelhet og Testbarhet.

Prinsipp:

Vedlikeholdbarhet skal prege all utvikling og forvaltning av løsninger. Komponenter og løsninger skal tilstrebe en teknisk kvalitet som forenkler vedlikehold og forvaltning gjennom hele livssyklusen til applikasjonen. En meget stor andel av IKT-relaterte kostnader i både privat og offentlig sektor er knyttet til (manglende) vedlikeholdbarhet, og det er avgjørende at vedlikeholdbarhet vektlegges i design, anskaffelse og utvikling av offentlige løsninger.

Tjenesteorientering (jf. avsnitt 3.2.1 i rapporten)

Dokumentet bør eksplisitt nevne at tjenesteorientering ikke nødvendigvis betyr at løsninger bygges med Web Services eller SOA-plattformer. Teknologier som REST, RSS, Atom og mikroformater bør også vektlegges. Det er en stor risiko for at dette prinsippet misbrukes som brekkstang for innføring av kostbare, omfattende og komplekse SOA-løsninger som innfører unødvendig kompleksitet, omfang og kost.

Fleksibilitet (jf. avsnitt 3.2.6 i rapporten)

Fleksibilitet anbefales moderert som prinsipp.

- Selv om målet er besnærende, er det i praksis både kostnadsdrivende samt vanskelig å gjennomføre og verifisere
- Utvikling av *"løsninger som er generiske og universelle slik at gjenbruk og flerbruk er mulig, til tross for at det eventuelt per tidspunkt for utviklingen ikke er aktuell problemstilling"* - krever ofte en stor grad av spekulasjon om framtidige behov. En slik tilnærming er erfaringsmessig blant de mest kostnadsdrivende i et IKT-prosjekt.
- Det bør derimot legges vekt på at utvikling av tjenester skal løse de konkrete behovene og ikke mer.
- For å oppnå målsetningene om fleksibilitet i praksis bør i stedet prinsipper som Enkelthet, Testbarhet og Vedlikeholdbarhet vektlegges. I praksis ligger ofte nøkkelen til fleksibilitet i disse prinsippene.

Skalerbarhet (jf. avsnitt 3.2.7 i rapporten)

Skalerbarhet anbefales moderert som prinsipp:

- Prinsippet innebærer en risiko for spekulasjon om fremtidig bruk, og erfaringsmessig stor risiko for at det spesifiseres unødvendig skalerbarhet. Dette er særdeles kostnadsdrivende og innfører kompleksitet som ofte er unødvendig.
- Skalerbarhet håndteres utviklingsmessig best som krav for hver enkelt løsning, ikke kun som et overordnet prinsipp.
- Det bør legges vekt på at utvikling av tjenester primært skal løse de konkrete behovene, mens man generelt for å oppnå målsetningene om skalerbarhet i i sterkere grad bør vektlegge prinsippene Enkelthet, Testbarhet og Vedlikeholdbarhet. I praksis ligger ofte nøkkelen til skalerbarhet nettopp i disse prinsippene.

Svar på spørsmål om arkitekturprinsippene:

- Finnes det arkitekturprinsipper som kan supplere de foreslåtte, og bør prinsippene presiseres/operasjonaliseres ytterligere?
Vi foreslår Enkelthet, Testbarhet og Vedlikeholdbarhet som nye prinsipper. Selv om det helt klart er behov for ytterligere presisering/operasjonalisering, er det viktigere å holde dokumentet kort og konsist. En mulig løsning er legge eksempler på de ulike prinsippene i vedlegg.
- Bør arkitekturprinsippene være helt eller delvis obligatoriske, eller bare veiledende?
På det nivået prinsippene ligger nå er det så stor tolkningsmulighet at spørsmålet ikke gir mening. Hvem skal avgjøre om prinsippene er fulgt eller ikke? Hva er konsekvensene; hva skal gjøres dersom de ikke er fulgt?
- Hvem bør omfattes av arkitekturprinsippene? Hele eller deler av statlig forvaltning? Statlig

forretningsvirksomhet? Kommunal- og fylkeskommunal sektor?

Alle

- Fra hvilket tidspunkt bør arkitekturprinsippene gjelde fra?

Fra de blir vedtatt Et spørsmål som må besvares er hva dette betyr for eksisterende løsninger og applikasjoner.

- Gitt en beslutning om å gjøre arkitekturprinsippene obligatoriske, hvilke konsekvenser får dette for etatens system- og tjenesteutvikling, herunder kvalitative, organisatoriske, juridiske, økonomiske eller teknologiske konsekvenser?

Ikke tatt stilling til dette i detalj. Men det er åpenbart at Statens standardavtaler trenger

tilpasninger ift blant annet eiendomsrett/intellectual property ift økende bruk av Fri Programvare på både infrastruktur-, applikasjons- og verktøy-/metodenivå.