



# **Proessorienterte løsninger i kommunal sektor**

**Fra forretningsbehov til kjørende prosess:  
Bruk av SOA og BPM i Bærum kommune**

**16. oktober 2008**



# BPM ”Proof of Concept” Samarbeidsprosjekt

Hilde Schjander Flugon  
IKT-avdelingen i Bærum kommune

Lars Flugstad  
Bærum kommunale bedrift Data

Mihaly Fekete  
CommITment AS

# Agenda

## Innledning

- Bakgrunn
- Hvorfor SOA?
- Hvorfor BPM?

## Tema

- Case
- Organisering
- Prosesskartlegging
- Tjenestemodell
- Etablering av SOA
- Implementering

## Erfaringer

- Prosjektet
- Implementering
- Prosess



COMMITMENT

# Bakgrunn



# Bærum kommune

## - noen faktaopplysninger

- Over 100.000 innbyggere
- 7.500 årsverk (herav 3.500 i PLO)
- 260 lokasjoner (skoler, sykehjem, med mer)
- MPLS nettverk med fiber til alle lokasjonene, svært god kapasitet - kan levere flere digitale tjenester
- 150 datasystemer (herav 20 Virksomhetssentrale)
- Undervisningsnett for 15.000 elever fordelt på 40 skoler med sentral administrasjon



COMMITMENT



# Offentlige føringer

- Stortingsmelding nr. 17 (2006-2007) - et informasjonssamfunn for alle
- eNorge 2009 – det digitale spranget
- eKommune 2009
- Program for digital kompetanse 2004 - 2008
- Bruk av åpne standarder og åpen kildekode
- Krav knyttet til annen tjenesteproduksjon



# Interne føringer

IKT skal være et strategisk virkemiddel for å nå kommunens mål

Med andre ord skal hovedfokus rettes mot kommunens tjenesteproduksjon og hvordan Bærum kommunes ledelse og resultatenheter skal nå sine mål gjennom **effektiv utnyttelse av IT- og kommunikasjonsløsninger**



# Virksomhetsarkitektur i BK

**Trenger en ny reguleringsplan for å kunne realisere føringene – satte fokus på virksomhetsarkitektur.**

**Gjennomførte i 2006 et forprosjekt sammen med Commitment for å**

- Kartlegge tjenesteproduksjon og applikasjoner
- Analyse av integrasjonsgrad mellom systemer
- Utarbeide nye målbilder for IKT infrastruktur basert på et betydelig sterkere tjenestefokus.

## **Konklusjon/oppgaver/innspill:**

- Etablere virksomhetsarkitektur
  - Metodikk / maler / roller
  - Forvaltningsregime
  - Arkitekturprinsipper
- Vurdere styringsmodell for IKT
- **Etablere tjenesteorientert arkitektur SOA**
- Kartlegge forretningsprosesser og fagsystem



# Hvorfor SOA?

- Muliggjør bruk av automatiske prosesser ved bl.a. ved utfylling av skjemaer på nett.
- Synkroniserer IT og organisasjon
- Forenkler og forener kommunikasjon mellom alle tekniske nivåer (brukergrensesnitt - prosesser - fagsystem)
- Gjenbrukbare tjenester
- Man kan innføre SOA gradvis i organisasjonen
- Forenkler samlet overvåknings og logging strategi
- Enhetlig IT strategi senker kostnader
- Muliggjør fokus på nyutvikling fremfor vedlikehold
- Kan bygges på eksisterende fysiske nettverk

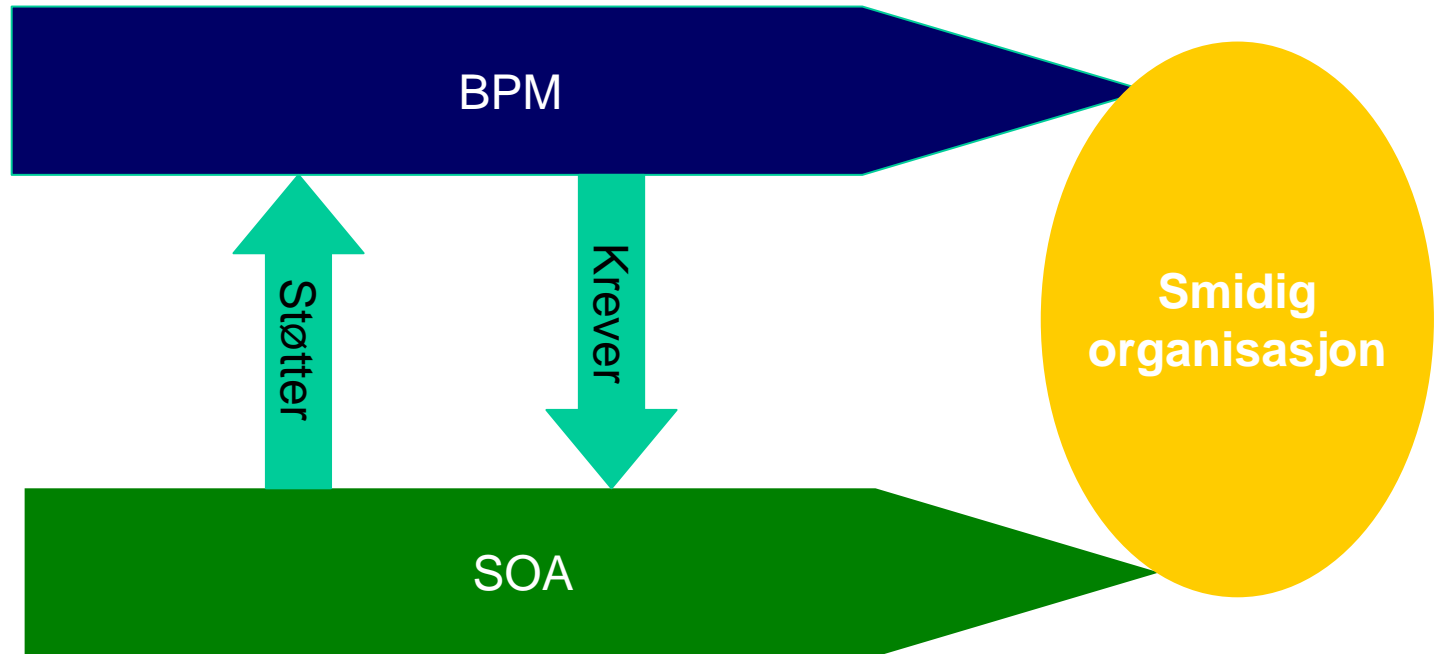
# Hvorfor BPM?

Mange av dagens IT løsninger møter ikke virksomhetens behov for fleksibilitet og effektivitet

- Industrialiserer kunnskapen som finnes i virksomheten
- Endringsdyktighet
- Sikrer at IT implementasjon større grad tilfredsstillir virksomhetens behov
- Reduserer redundans av funksjonalitet tvers gjennom ulike IT systemer ved økt standardisering
- Sikrer presis nok prosess beskrivelse for å unngå mistolkning ved implementering ...
- Eller gir presis nok prosess beskrivelse for maskinelt eksekvering (BPMS)

# BPM og SOA relasjon

- To initiativer med et felles mål
- BPMS orkestrer tjenester som tilgjengelig gjort av SOA





COMMITMENT

# Prosjektet



# Mandat til prosjekt @SOA

@SOA har ansvaret for å velge, designe og implementere SOA-plattform og tilrettelegge for eBærum slik at kommunen kan realisere selvbetjente tjenester på nett. Modell for flere tjenester i 2009

## Forsknings- og utviklingsprosjekt for å

1. Bestemme og etablere SOA infrastruktur (ESB, BPMS med mer)
2. Kvalitetssikre konsept og løsninger ved å implementere automatisert løsning for SFO innmelding
3. Utarbeide metodikk, maler, retningslinjer med mer (forvaltning av tjenester realisert på SOA-plattform)
4. Utarbeide arkitekturprinsipper (navnestandarder, sikkerhet, åpne standarder)



# Organisering

- CommITment som deltager i SESAM-prosjektet foreslo å gjennomføre BPM Proof of Concept i Bærum kommune
- Prosjektet eies av kommunens IKT-avdeling, som er en strategisk enhet
- Teknisk gjennomføring ivaretatt av systemavdeling hos kommunens interne driftsleverandør (BKB Data)
- Proses design og implementering utført av resurser fra CommITment

# Mål for Proof of Concept

- Etablere sikker pålogging med Kobling til Min Side
- Implementere SOA-plattformen
- Designe og implementere automatisert løsning for SFO-innmelding så nært opp til produksjon som mulig
- Etablere arkitektur og forvaltningsprinsipper for bruk av SOA-plattform



COMMITMENT

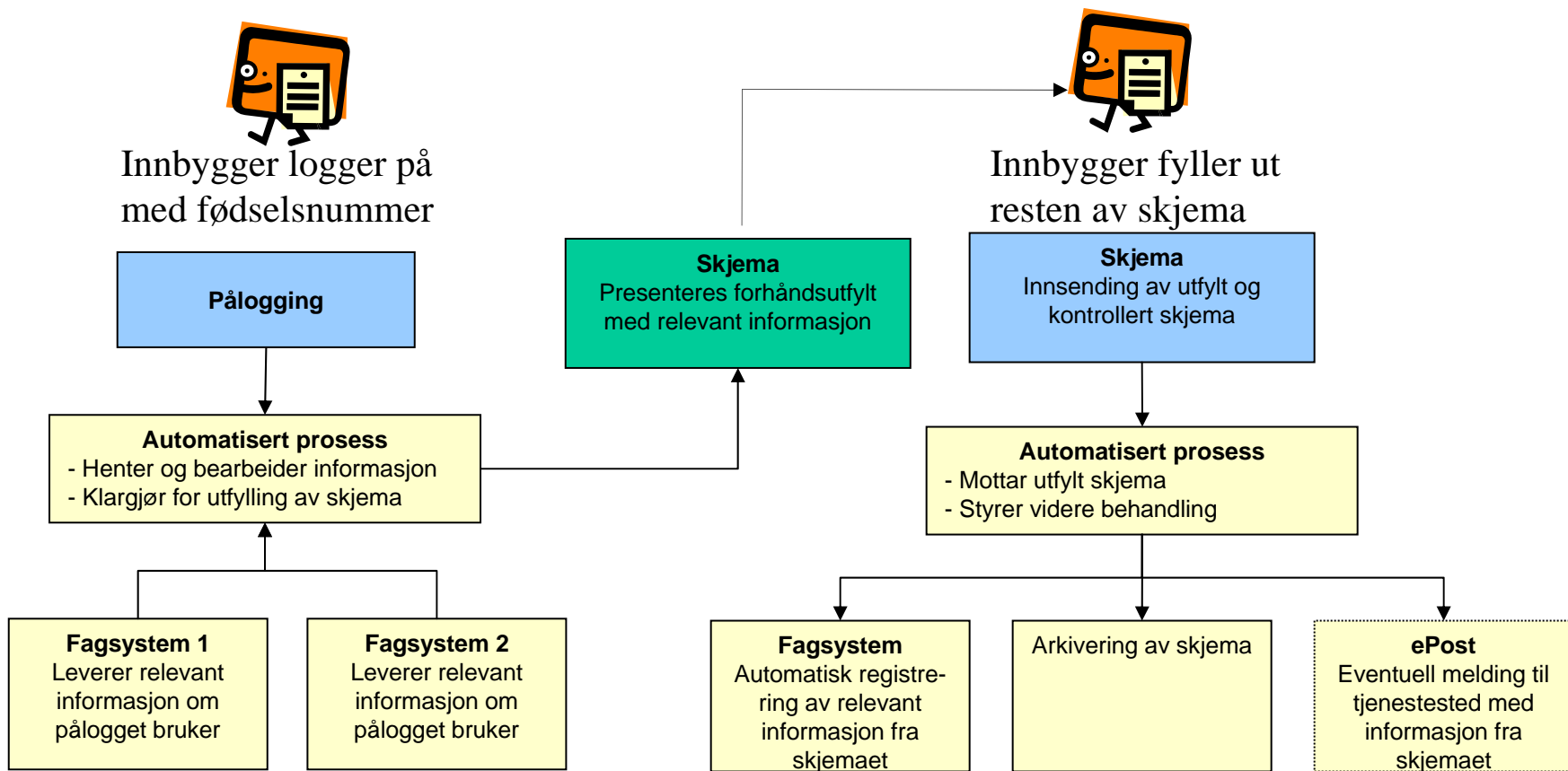


# Case

## Helautomatisert SFO innmelding



# Elektronisk arbeidsprosess fra skjema via integrasjonslag

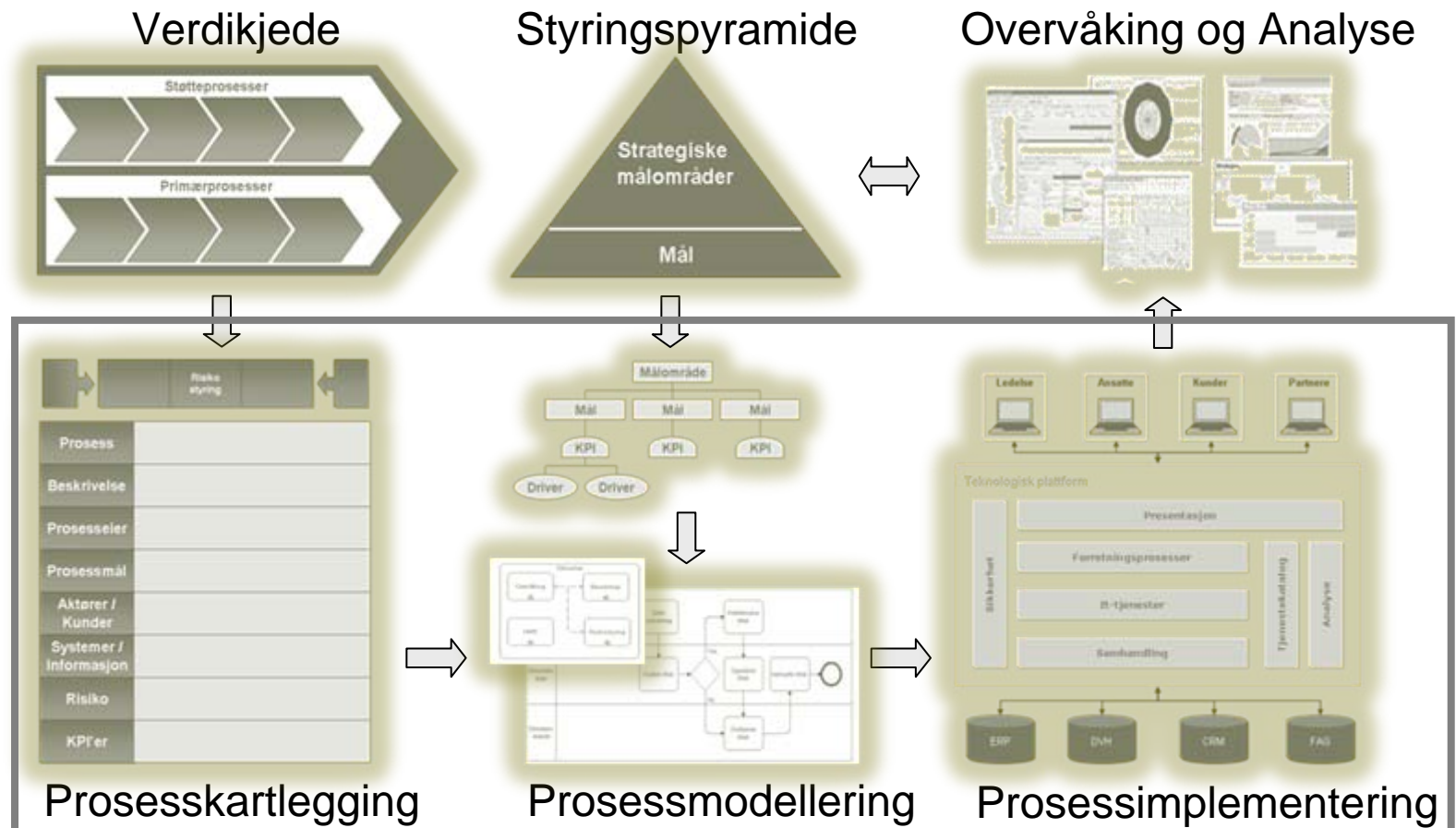


# Etablere SFO-løsning

- Kartlegge eksisterende prosess
- Klargjøre regelverk
- Tegne ønsket prosess
- Beskrive den funksjonelle tjenesten
- Identifisere nødvendige tjenester og mulighet for gjenbruk
- Lage eventuelle nye komponenter, modellere prosessen
- Teste

# Prosessforvaltning

Prosesstankegangen vil være gjennomgående i hele virksomheten og endre dagens utviklingsprosesser





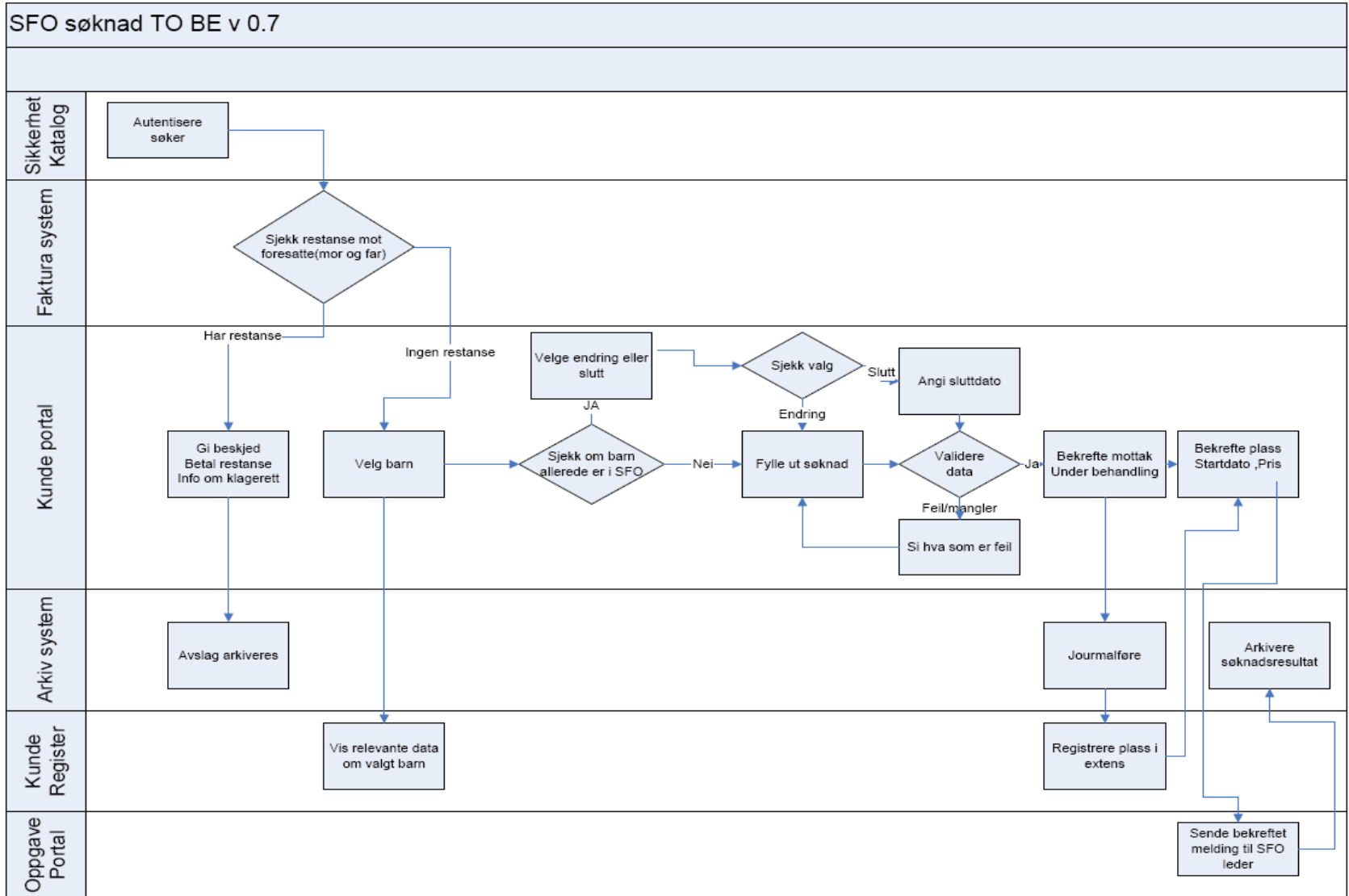
# Forutsetninger for TO-BE prosessdesign

- Innmeldingsskjema skal tilgjengeliggjøres via kommunens webside
- Forenklet brukergrensesnitt – mer enn et tomt skjema
- Fullautomatisert behandling



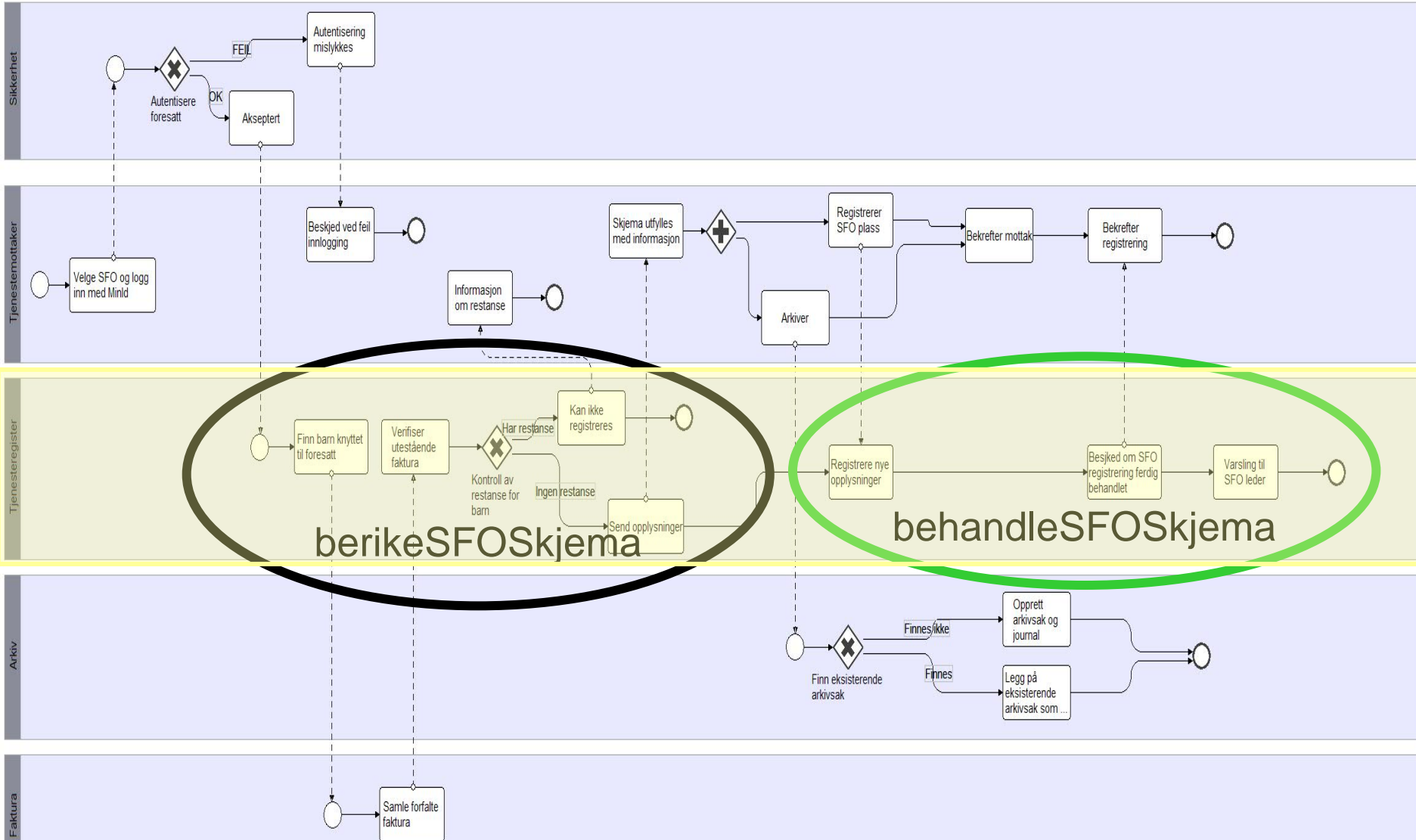


# TO-BE overordnet prosessmodell





## Funksjonell prosessmodell





# Dokumentering av tjenester

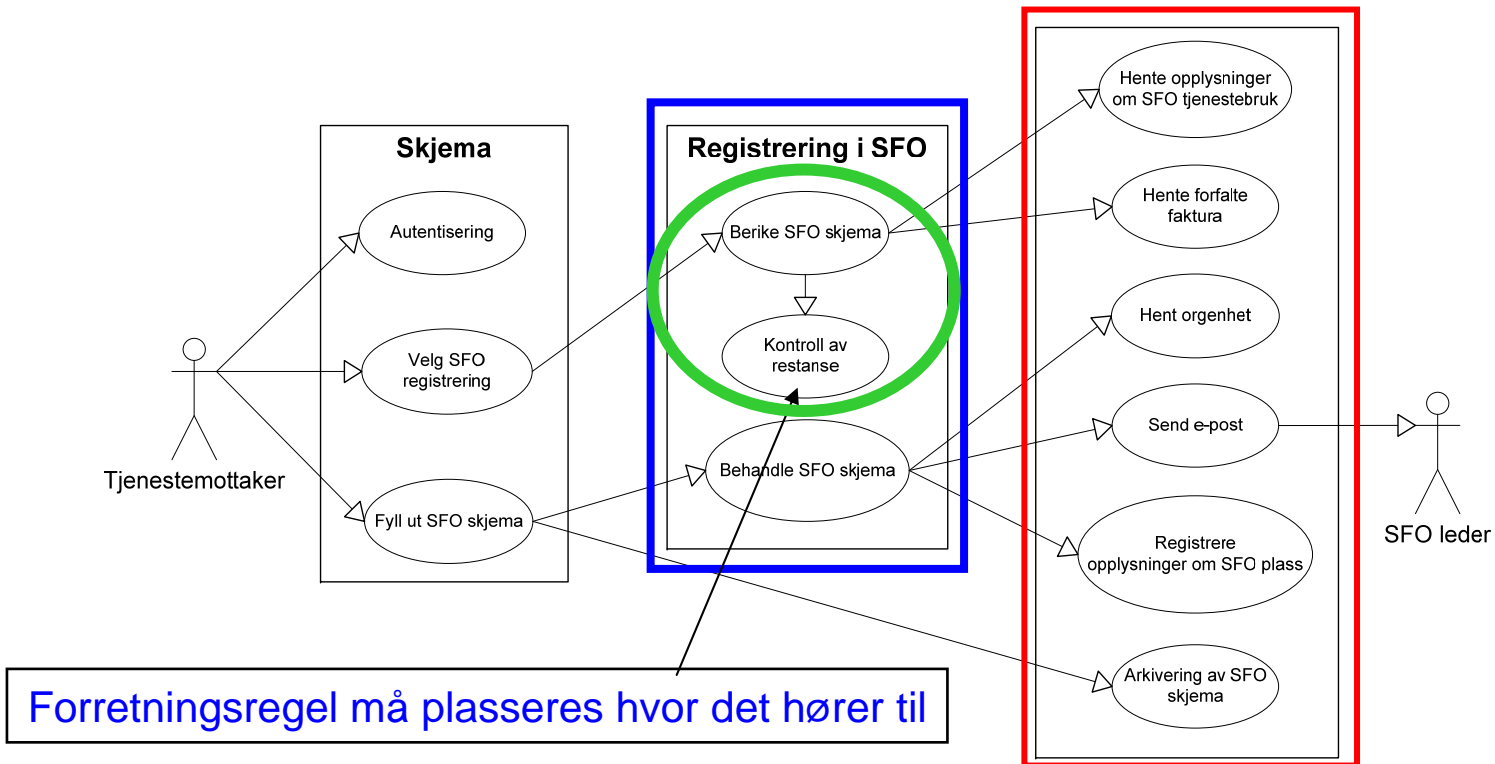
Prosess modell på Forretningsnivå blir kravspesifikasjon og ikke en implementerbar design spesifikasjon

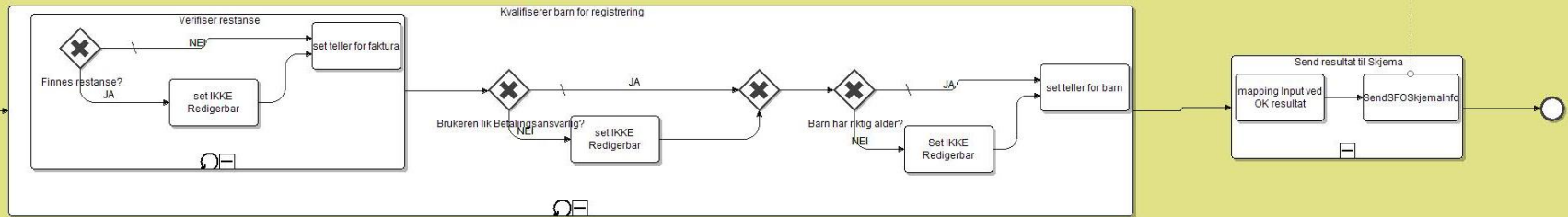
Alle tjenester spesifiseres i 2 nivåer

- **Funksjonell Tjenestebeskrivelse**  
hensikt, eier, beskrivelse av tjeneste, funksjonell [prosess] design, egenskaper, bruksmønstre, funksjonelle avhengigheter og -avvikshåndtering
- **Tjeneste Arkitekturbeskrivelse**  
implementasjon design (BPMN og/eller Use case), informasjon modell, interaksjon med klient (metoder), krav til drift og sikkerhet, teknisk avvikshåndtering, avhengigheter til valgt teknologi, SLA avtale

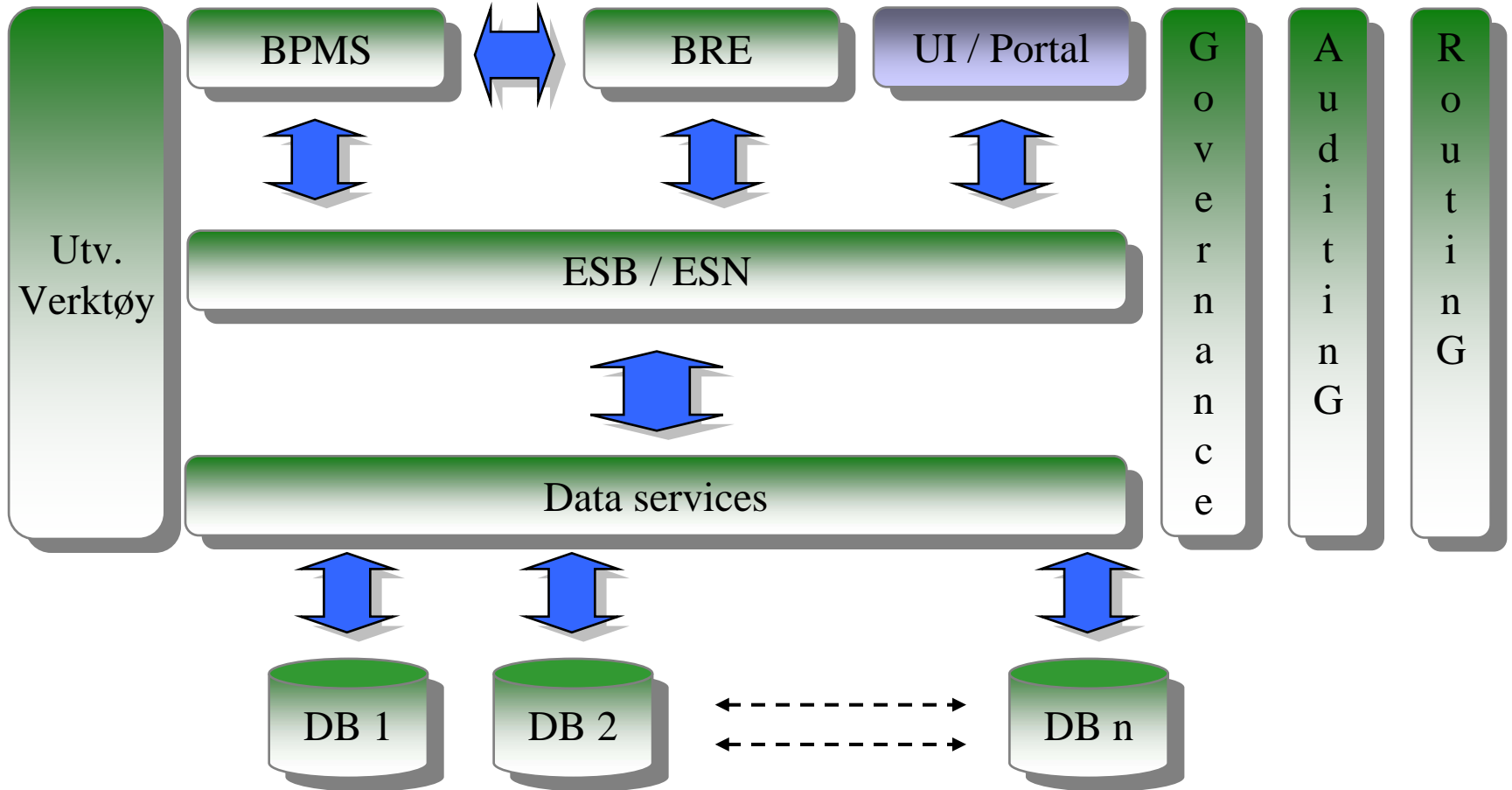
# Identifisere tjenester

Identifisere og beskrive tjenester som skal brukes av prosessen basert på Use case modellering





# SOArkitektur





COMMITMENT

# Kriterier for valg av komponenter

- Kjøpevare / Open Source
- Storingsmelding nr. 17 (2006-2007)
- Ikke versjon 1.0 eller bare beta release
- Mange installasjoner
- Aktivt community
- Mulighet for bistand / kurs i Norge
- Skal primært bruke åpne standarder i grensesnittene / tilgangspunktene
- Støtte for multiple og standard kommunikasjonsprotokoller (HTTP etc.)
- Multiple plattformer (Linux, Windows)
- Ukompliserte /effektive rammeverk
- Støtte for ”standard” sikkerhetsregimer (LDAP, Acegi, WS-S etc.)
- Enkelt å programmatisk utvide / tilpasse komponenter
- Programmeringsspråk – Intet moment

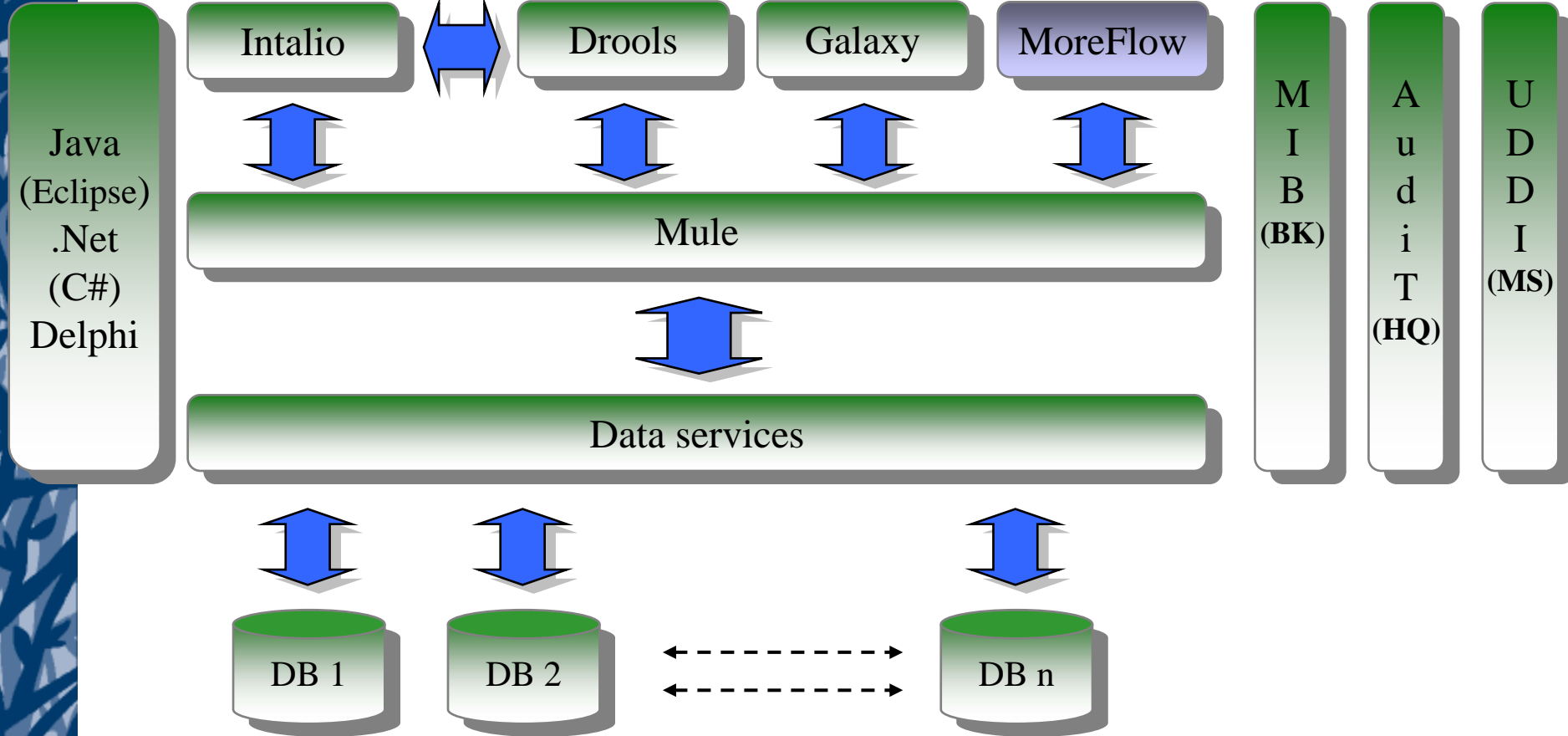
# Kriterier for valg av komponenter

## Bærum spesifikt

- Enkelt integrasjon mot eksisterende fagsystemer og nye integrasjonspunkter (WS, API, direkte DB oppslag, fillesning etc.)
- PoC => Infrastruktur (Ingen utskifting av komponenter etablert under PoC)
- Governance skal helst være et tillegg til eksisterende governance løsning eller fungere i parallell
- Auditing skal helst fungere i parallell eller komplementere eksisterende ”rapportering og monitorering”
- Utviklingsverktøy; Delphi, C# eller Java



# BK - SOArkitektur





# Delvis introduserte komponenter

- Galaxy
  - Mangler fortsatt mulighet for å lagre forvaltningsinformasjonen i egen database / predefinert destinasjon
  - Kan ikke assosiere properties-filer med konfigurasjonsfiler
- HQ
  - Er utenfor scopet av selve PoC'en, men testes for fremtidig bruk





COMMITMENT

# Erfaringer

# Bestem eierskap og ansvar

## Erfaring

- Hvem bør eier tjenesten?
  - Den som er ansvarlig for prosessens sluttresultat og derfor motivert for å gjennomføre kontinuerlig effektivisering.
  - Eierskap må også innebære full og detaljert forståelse av tjenesten med tilhørende prosess
- Viktig å avklare eierskap til regelverket
  - Ingen rom for ”personlige tolkninger” av regelverk i automatisert prosess
  - Automatisering betyr ofte endrede oppgaver for folk
- Hvem har ansvar for forvaltning av prosessen og regelverket etter implementasjon?



# Gjennomføring

## Erfaring

- Samhandling mellom virksomhet og teknisk er utfordrende, men helt nødvendig
- Mye tid brukt til å vente på leveranser fra leverandører, eller gi gjentatte tilbakemeldinger
- Vanskelig å implementere infrastruktur, tjeneste og rammeverk samtidig
- Mye mer teknisk prosjekt enn antatt
- Utfordrende å systematisere dokumentasjon
- Ting tar tid!



# Noe erfaringer med prosess modellering

## Erfaring

### 3-lag prosess modell

- Forretning : kun aktør og flyt av aktiviteter (kan beskrives med hvilken som helst notasjon)
- Funksjonell : beslutningspunkter, tjenestekall, subprosesser og informasjonsobjekter (beskrives gjerne med BPMN)
- Implementering : Exception, Compensation, Timer, Data mapping, Synkronitet / A-synkronitet (beskrives med BPMN)

### Del ut gjenbrukbar funksjonalitet i

- Subprocess
- Egen prosess/tjeneste som er tilgjengelig som Web service
- En typisk tilfelle er regler ved "long running" prosesser

**Tjeneste katalog løsning øker effektiv gjenbruk av tjenester i prosessen**

**Bruk "Contract-first" prinsipp og begynn prosess implementering mot "mock-up" tjenester**

**Ferdigdefinert felles data modell er avgjørende for effektiv prosess implementering**

# Contract first vs Contract last

## Erfaring

- Ved "Contract last" overlater man sin tjenestens skjebne til et rammeverk.
- Tjenesten utvikles ut ifra ferdig definert "Contract" både hos tjener og klient
- Web services velges hovedsakelig fordi man ønsker et klart definert grensesnitt som er implementasjon uavhengig.
- WSDL og XSD er ikke spesifikasjon men realiseringen av spesifikasjonen

# Semantikk

## Erfaring

- Forretning ⇔ Teknisk

- Semantikk, ikke bare tekniske begreper, men GISP gjelder også mennesker!!
- Annen type bestiller og utfører kompetanse
- Tekniske beskrivelser, mer fokus på XSD, XML etc. enn ”vanlige” utviklingsprosjekt
- Organisasjon og organisering
- Prosjektet vurderte muligheten for å benytte allerede eksisterende klassifiseringssystemer, men kom til at dette ville kreve for mye ressurser av en PoC som var rettet mot kompetanse og erfaringsoppbygging.
- Et felles samtaleunivers ble bygget opp med begreper mer tilhørende den forretningsorienterte siden enn den tekniske.

- Kommunikasjon mellom forretning og teknisk

- Beskrevet i to typer av dokumentasjon:
  - Funksjonell tjeneste dokumentasjon
  - Teknisk tjenstedokumentasjon
- Kvalitetssikring av dokumentene ble tillagt en spesifikk rolle

# Kontekst

## Erfaring

Som i vanlig lag orientert (n-tier) arkitektur ble det designet to typer kontekster:

- Innkontekst
  - KorrelasjonsID
  - Annen prosess spesifikk metainformasjon
- Utkontekst
  - Resultat av forespørsler
  - KorrelasjonsID
  - Statusinformasjon (Feilkoder og beskrivelser)
    - Forretningsorienterte feil
    - Teknisk orienterte feil

I tillegg fanges abnormale feil (Exceptions) som sendes til eget feil management system. Til den kallende prosess sendes det en melding om teknisk feil i et objekt av typen utkontekst.

# Erfaringer generelt

## (Hovedpunkter)

### Modenhet av open source programvare er veldig variabel

Mule ↔ Intalio

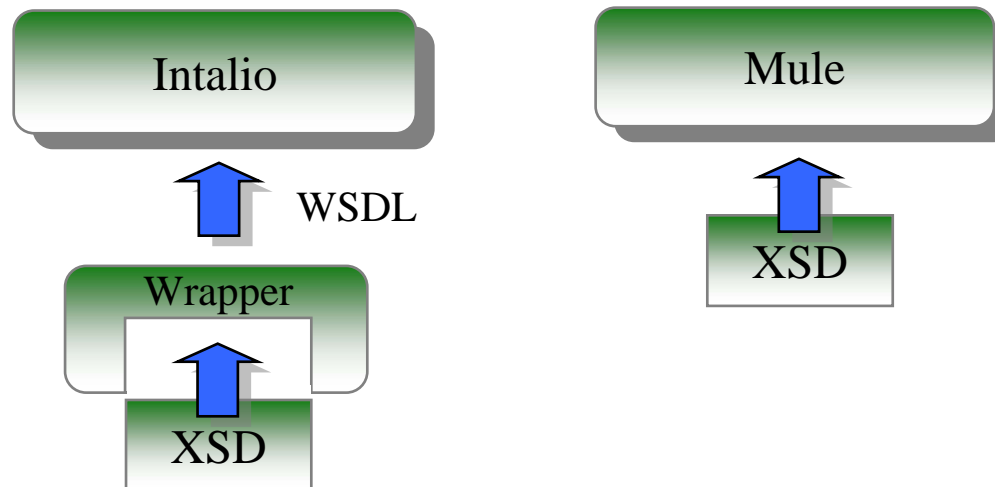
- Axis 1.x fungerer dårlig i grensesnittet mot Intalio
- Axis 2.x er ikke støttet i verken Mule 1.x serien eller tilhørende Mule IDE "Out of the box" pr. dags dato.
- XFire fungerer mest optimalt, men skaper problemer da den hele tiden legger til "ArrayOf" på alle listetyper definert i XSD
- Intalio med sin Axis2 er orientert mot "Contract first" WSDL utvikling, mens Mule med XFire er mer XSD orientert.



# Erfaringer generelt

## (Hovedpunkter)

XFire eksponerer komplette WSDL'er noe Axis2 har problemer med å tolke hvis flere XFire eksponerte tjenester har flere felles XSD schema definisjoner. Intalio svarer med at en har to identiske typer i samme namespace. For å kompensere for ovennevnte problem ble det laget "wrappere" rundt XSD'er som Intalio kan importere slik at utvikling kan skje parallelt på både Intalio og Mule.



(NB: Mule 2.x med bruk av CXF løser denne problemstillingen da den også tar utgangspunkt i WSDL)



# Erfaringer generelt

## (Hovedpunkter)

### Mule ⇔ Fagsystemer

- Mule med XFire mot Delphi 2007 WS og .Net 2.0 WS fungerer uten problemer når Mule er satt opp som proxy.
- Ved alle situasjoner hvor det skal skje en eller annen form for transformasjon standardiserte vi på CXF (ver. 2.0.5) i tillegg til en egenutviklet komponent.
- Et alternativ til punktet over er å utvikle egne transformatorer for XML binding (JiBX).
- For å generere C# klasser i .Net (xsd.exe) er det nødvendig å lage egne element definisjoner da klassegeneratoren til .Net ikke er beregnet på å generere klasser kun ut fra complex og simple types definisjoner.

# Erfaringer generelt

## (Leverandører)

- Løsninger levert av mange leverandører er ofte teknologiske løsninger med foreldet arkitektur. De er ennå ikke modne for leveranser med WS orientert grensesnitt.
- Forretningsstrategien er i stor grad salg av ”silo” produkter og ikke tjeneste orientert
- Lite fleksible for nye typer leveranser som kreves i en SOA.



# Ikke-tekniske utfordringer fremover

## JUSS !!!

Vanligvis er applikasjonsdomenet sterk knyttet til en eller flere applikasjoner med tilhørende data og eies samt forvaltes av en systemeier. Fra en "Silo" tankegang til SOA vil fagsystemer som eksponerer tjenester benyttes av flere tjenestebrukere...

### Utfordringer:

- Hvem skal betale for utviklingskostnadene relatert til eksponering av tjenester?
  - Systemeier?
  - De som ønsker at systemet eller deler av systemet skal eksponeres og tilbys som tjenester?
  - Begge?
- Drift
  - Hvem er ansvarlig for drift av tjenesten?
  - Krav til oppetid? Forventes det 24/7 oppetid?
  - SLA kontrakter; Er de i eksisterende kontrakter dekkende nok ?
- Forvaltning og videreutvikling
  - Hvem er ansvarlig for forvaltning og videreutvikling av tjenesten?



COMMITMENT

# Fremtiden

- Slutføring test og implementering av Galaxy som SOA governance i parallell med eksisterende forvaltningsløsninger i kommunen.
- Slutføring av HQ i parallell med Galaxy. Det legges opp til at HQ skal benyttes som auditing verktøy for både SOA løsninger og eksisterende løsninger der det er hensiktsmessig.
- Drools er prosjektert for testing desember 2008.
- UDDI –Q1 2009 (bruk av både statiske og dynamiske WSDL'er).
- Bedre organisasjons assemblering i prosjektstart med utgangspunkt i OIO eller lignende metodikk.
- Videreutvikle virksomhetsarkitektur og infrastruktur i hele kommunen.
- Kartlegge applikasjonsdomener og potensielle tjenester felles for hele kommunen



COMMITMENT



## Erfaringer med fri programvare

- Fri programvare er ikke gratis, men koster mindre!
- Fri programvare til bruk i sentrale løsninger og i infrastruktur medfører
  - Åpenhet, standardisering og mye tilgjengelig kompetanse
  - Må ha betalbare tilleggsprodukter utover community-versjon – og det finnes masse valg
  - Må ha avtale for support- og versjonsgarantier
- Applikasjoner basert på fri programvare
  - Finnes få løsninger for kommunal anvendelse i dag



## Konklusjon fri programvare

- Vi er svært godt fornøyd med valgene som stort sett fungerer som forventet. Selv om begge bruker samme åpne standarder er det noen tilpasninger for å få til samhandling mellom de ulike lagene.
- God og tilgjengelig norsk/lokal kompetanse/support/kursing på MULE i tillegg til på nettet
- Litt mindre for Intalio, men der er det derimot svært mange kunder

# Bærum kommune

Og til dere som tenker i samme retning men ennå ikke ”tar sjansen” og fortsatt er i tvil:

- Bærum kommune ønsker å dele sin kompetanse/erfaringer og sine løsninger med resten av offentlig forvaltning
- Bærum kommune ønsker å bidra til større konkurranse i markedet ved å satse på utstrakt bruk av fri programvare og åpne standarder.





# Prosjekt @SOA

## Noen lenker

<http://www.baerum.kommune.no>

<http://www.commitment.no>

<http://www.mulesource.org/>

<http://www.intalio.com/>

## Kontaktinfo

### **Bærum kommunale bedrift Data**

[lars.flugstad@baerum.kommune.no](mailto:lars.flugstad@baerum.kommune.no)

[oystein.aanrud@baerum.kommune.no](mailto:oystein.aanrud@baerum.kommune.no)

### **Bærum kommune IKT-avdelingen**

[hilde.flugon@baerum.kommune.no](mailto:hilde.flugon@baerum.kommune.no)

### **CommITment AS**

[mihaly.fekete@commitment.no](mailto:mihaly.fekete@commitment.no)