

Virksomhetsarkitektur i Enova

En virksomhetsarkitektur skal legge til rette for at forretning, organisasjon, arbeidsprosesser, teknologi og IT-systemer skal utvikles i takt og fungere sammen som en helhet. Dette har aktualisert seg i Enova over de siste par årene, da Enova er i gang med en kraftig satsning på IT og hvordan IT kan bidra i forretningsutviklingen. Vår organisasjon har gradvis modnet seg til å venne seg til å se IT og forretningsutvikling i sammenheng og flere av våre medarbeidere har fått øket sin kompetanse om prinsippene for virksomhetsarkitektur, uavhengig av hvor og på hvilket nivå i organiseringen av forretningen de hører hjemme.

Vi har valgt å legge rammeverket TOGAF til grunn for å beskrive vår virksomhetsarkitektur, og har utarbeidet en *arkitekturvisjon*. Denne beskriver framtidig målbilde for Enovas virksomhetsarkitektur og utdyper arkitekturprinsippene vi har valgt. Vi har også beskrevet benyttede rammeverk, samt roller, organisering og prosesser for arkitekturarbeid i Enova.

Dette notatet er først om fremt rettet mot den interne ledelsen i Enova, og er en proof of concept på målarkitektur og hvordan den kan bygges opp for å møte forretningsens behov. Vi velger å offentliggjøre notatet for å vise hvordan vi tenker, hvordan vi kommuniserer rundt dette internt, og hvilke begreper vi bruker. Vi tar gjerne imot innspill på justeringer av målarkitekturen på dialogmøtet.

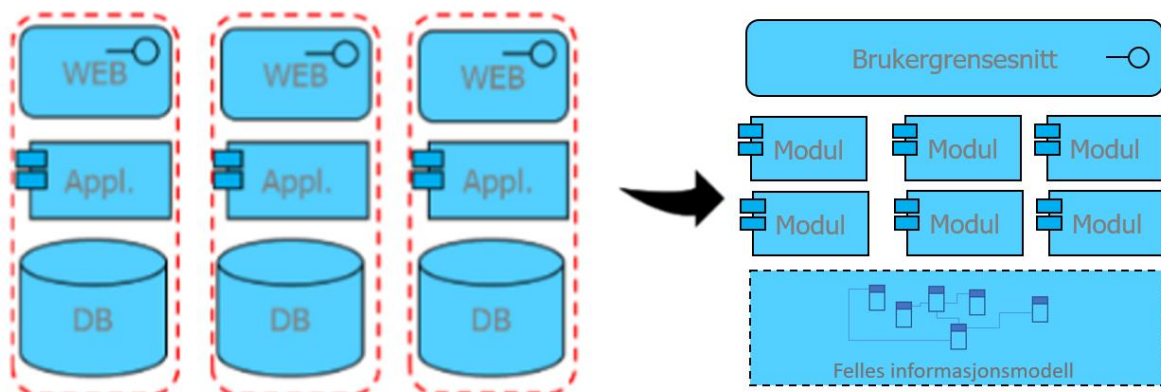
Virksomhetsarkitektur

Rapport etter prosjektets fase 3 pr. 20. november 2018

- Målarkitektur
 - PoC for målarkitektur basert på Dynamics 365
 - Modellering av informasjonsmodell i CDM, CDS-integrasjon med eksisterende datakilder og forsøk på rapportering etter markedsendringsmål. (august/september)
 - Etter gjennomført PoC ble resultatet verifisert og kommentert av Gartner. Denne oppdaterte versjonen av rapporten inkluderer innspill og forslag til forbedringer fra denne gjennomgangen.

Hensikt

Virksomhetsarkitekturprosjektets fase 2 skisserer en målarkitektur som vist under. Hensikten med aktiviteten beskrevet i dette notatet er å verifisere om dette er en vei å gå for Enova, om markedet understøtter vår tankegang, og om det praktisk er mulig å gjennomføre. Dersom Enova vedtar at dette er vei å gå, blir dette et viktig underlag til anskaffelser fremover, både med tanke på nye fagsystemer og andre tilgrensende IT-systemer.



Figur 1: Enovas målarkitektur

Hovedhensikten med målarkitekturen er å skape nødvendig fleksibilitet i fremtidig IT-arkitektur. Vi ønsker å komme vekk fra monolittiske applikasjoner til mindre moduler som i større grad kommuniserer med hverandre basert på en felles forvaltet informasjonsmodell. Dersom det er tydelig definerte grensesnitt mellom modulene kan vi endre eller

erstatte en modul ved behov uten at det får store konsekvenser for de resterende. Dette gjør det også enklere å innføre en ny modul som for eksempel kan nyttiggjøre seg av gode «kundedata» og andre sentrale entiteter i Enova (virkemiddel, sektor, kunde osv). I et ideelt fremtidsbilde har vi med andre ord en IT-arkitektur som består av moduler som er «løst koblet» til resten av arkitekturen, hver modul kan dermed forvaltes og videreutvikles i eget tempo og fordi disse er løst koblet vil det ikke være nødvendig å gjøre endringer i noen av de andre, ref. «Wardley mapping». På den måten kan vi sikre at Enova hurtig kan innføre ny eller endre eksisterende funksjonalitet basert på forretningsens behov.

Innramming

Vi har valgt å gjøre PoC på Microsoft Dynamics 365 CDM/CDS, men hovedformålet har vært å verifisere at målarkitekturen er realiserbar. For å skape en motvekt til Microsoft, har vi gitt Oracle vår målarkitektur og fått dem til å presentere sin plattform/portefølje i denne konteksten. En kort oppsummering av dette er med i rapporten.

Innledning

I Microsoft sin produktportefølje, er det Dynamics 365 med Common Data Model (CDM) og Common Data Services (CDS) som kan brukes for å realisere deler av målarkitekturen. Denne aktiviteten i prosjektets fase 3, kjører en Proof of Concept (PoC) på CDM/CDS. Der ønsker vi å få verifisert at vi kan modellere vår informasjonsmodell inn i CDM og i tillegg koble oss til eksisterende datakilder i dagens system (og noen Excel-ark) med CDS for å fylle databasen med data.

Proof of Concept (PoC) ble gjennomført primo september 2018 med hjelp av Prosesspilotene. Avtalt leveranse i delprosjektet var "Deler av Enovas databasemodell skal mappes og dokumenteres mot CDS. CDS tilpasses i samsvar med denne dokumentasjonen. Modellen er ikke uttømmende for alle felter, men så langt som mulig innen de rammer som er satt til POC. Hensikten er å se om CDS kan være bærer av all data for Enova."

Hva har vi lært?

En moderne og fleksibel IT-arkitektur kan ikke alene baseres på en felles informasjonsmodell, det er nødvendig med kompletterende infrastruktur og designprinsipper for å støtte opp under informasjonsflyt mellom ulike lagringsløsninger, 3. parts applikasjoner og egenutviklede moduler/applikasjoner. Gjennom PoCen har vi berørt og diskutert flere mulige konsepter og komponenter som kan være aktuelle for Enova. De mest aktuelle i så måte er:

Felles informasjonsmodell

En felles informasjonsmodell betyr ikke at alle data for alle applikasjoner skal ligge i en og samme fysiske database. En felles informasjonsmodell innebærer derimot at Enovas sterke entiteter (med tilhørende attributter) er vel definert, forankret og tydelig angir hvilket system som er master for hver enkel entitet. Entiteter som er felles for en eller flere moduler/applikasjoner skal kunne utveksles via en buss, og ikke som punkt til punkt integrasjoner.

Microtjenester

Microtjenester er egentlig ikke noe nytt (og er egentlig en variant av SOA) og handler om å bryte opp tjenester i små deler og gjøre disse tilgjengelig både internt og eksternt (for eksempel kan microtjenesten «Ny kunde» opprette en ny kunde i CDM). Å bryte opp en IT-arkitektur i små microtjenester var tidligere for krevende, men introduksjonen av store skalerbare (og relativt billige) skytjenester har nå aktualisert designprinsippet. Microtjenester er ofte realisert med veldig enkle protokoller (REST Api) og veldig spesialiserte. (Fra Unix «Do one thing, and do it well»).

Minitjenester

En minitjeneste er ikke noe annet enn en pragmatisk versjon av en microtjeneste. Det vil si at prinsippet om å bryte opp arkitekturen i mindre uavhengige («løst koblete») deler er det samme, forskjellen ligger i hvordan man deler opp funksjonaliteten. En minitjeneste kan betraktes som en samling med microtjenester som til sammen utfører en forretningsfunksjon. «Saksbehandle søknad» kan være et eksempel på en minitjeneste for Enova. En tredjepartsapplikasjon, som for eksempel dagens Basware, kan også betraktes som en minitjeneste, dette fordrer derimot at 3. parts applikasjonen har grensesnitt som muliggjør kommunikasjon med moderne hendelsesbusser.

Funksjoner (Function as a Service, FaaS)

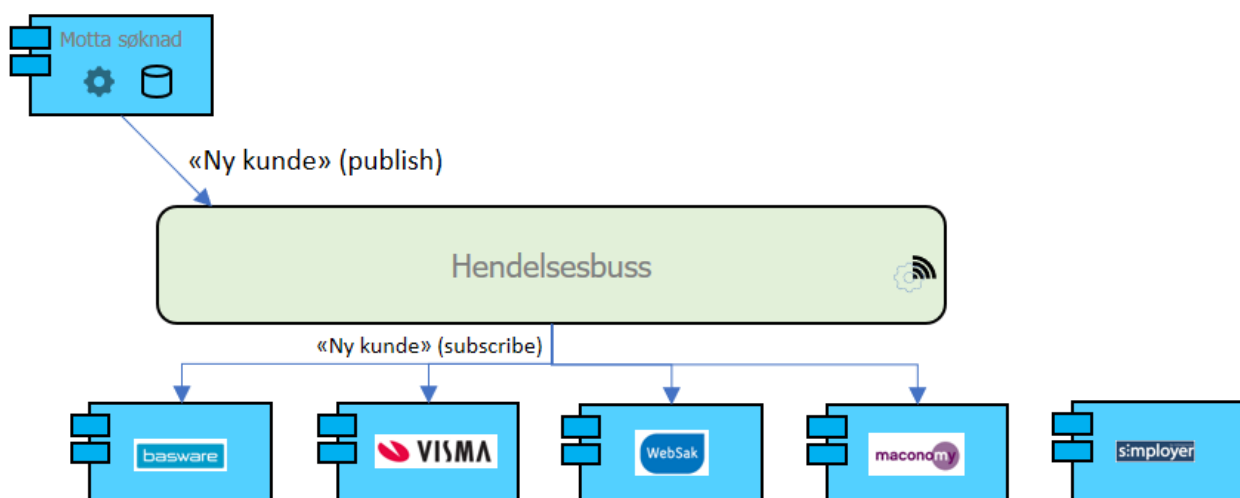
Funksjoner kan sees på som en samling av tjenester som til sammen utfører en større operasjon. En funksjon har en «trigger» som starter jobben, nødvendige systemressurser blir så aktivert for å utføre funksjonen. Fordelen her (i forhold til Platform as a Service) er at den ikke krever at en server har denne tjenesten kontinuerlig oppe, systemressurser aktiveres bare når funksjonen trigges og eksekveres. Dette er også populært kalt «Serverless computing».

Hendelsesbuss (Service bus)

Bruk av microtjenester, minitjenester og funksjoner genererer en hel rekke meldinger mellom applikasjoner og databaser i en IT-arkitektur. En tjenestebuss bistår med overvåking av denne meldingstrafikken og sørger for at alle applikasjoner/databaser mottar/sender all informasjon som kreves til riktig tid og i riktig rekkefølge. Den vil også sørge for å lage en meldingskø til systemer som av en eller annen grunn ikke er tilgjengelige når den opprinnelige meldingen sendes. I tillegg tilbyr gjerne en tjenestebuss funksjonalitet for publisering/abonnering på meldinger som sendes gjennom.

Et tenkt eksempel:

En mulig egenutviklet minitjeneste i Enovas arkitektur kan være tjenesten «Motta søknad». Denne tjenesten har ansvaret for å håndtere en søknad fra det øyeblikk den er sendt inn av kunde til den er videresendt til riktig saksbehandlingstjeneste (privat eller næring). En av oppgavene tjenesten gjør er å sjekke om kunden allerede eksisterer i Enova eller om det er en ny kunde. Dersom kunden er ny sender tjenesten ut en melding til hendelsesbussen av type «Ny kunde». Meldingen inneholder da alle data tjenesten har mottatt fra søker (og evt andre oppslagsmuligheter).



Minitjenester (3. parts applikasjoner i dette tilfellet) som trenger informasjon om alle søkere abonnerer på denne meldingstypen («Ny kunde») mottar meldingen og kvitterer at den er mottatt. På denne måten kan vi sikre at data kommer inn en plass en gang, og deretter flyter ut til alle andre systemer som trenger samme informasjon.

Dersom alle integrasjoner følger dette mønsteret vil vi oppnå løst koblede applikasjoner, som igjen innebærer at vi for eksempel kan erstatte økonomisystemet og «koble inn» et annet bare ved å sørge for at meldingene inn/ut av hendelsesbussen er like.

API Gateway

På samme måte som det er formålstjenlig å etablere løse koblinger mellom tjenester og applikasjoner, er det en fordel at all kommunikasjon fra brukergrensesnitt og inn mot bakenforliggende funksjonalitet gjøres gjennom en API Gateway. Med en gateway kan man enkelt gjøre endringer både i grensesnitt og interne tjenester fordi man kun forholder seg til de definerte grensesnittene både på inn- og utsiden av gateway. En gateway kan også håndtere følgende funksjonalitet på vegne av alle moduler som tilbys utad:

- Autentisering
- Lastbalansering
- Brannmur (web)
- Logg og overvåkning
- Styring av pågang (throttling)
- SSL sertifikater

For 3. parts applikasjoner, som Enova enten installerer selv eller leier i sky, vil den beste løsningen være å benytte standard grensesnitt som kommer med applikasjonen.

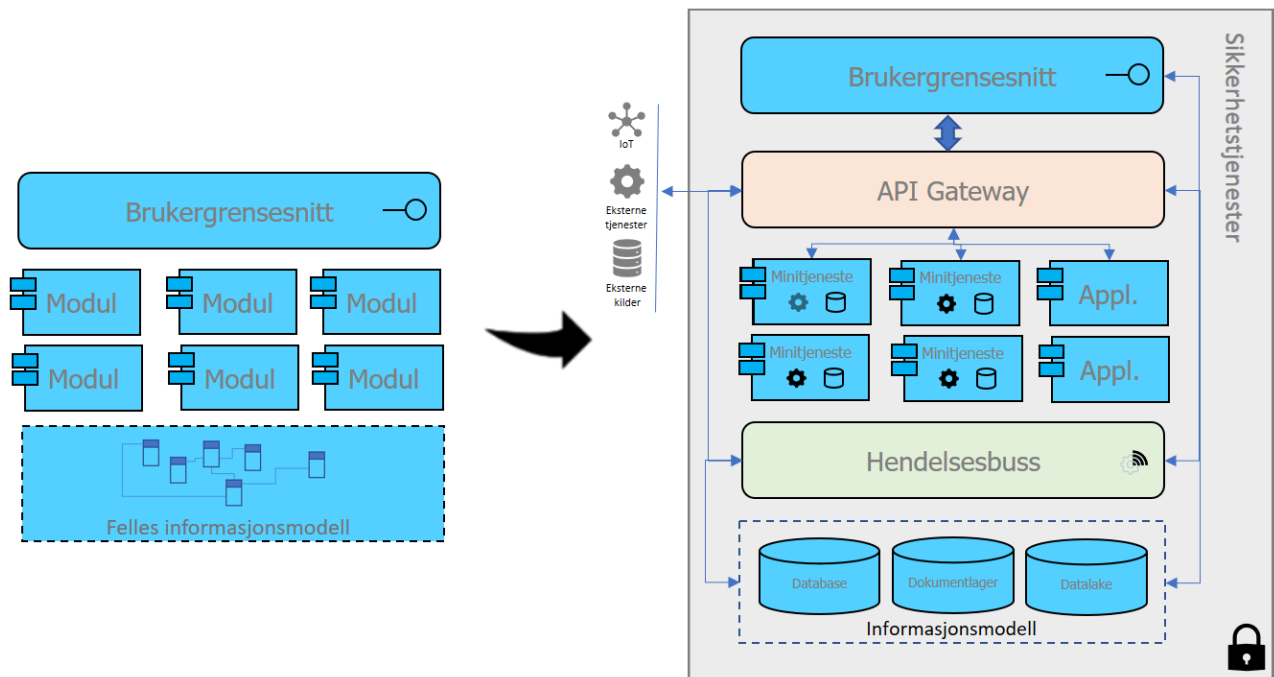
Lagringsmedium

En moderne arkitektur bør også kunne være fleksibel med hensyn på å tilby ulike typer lagringsmedium tilpasset egenskapene ved dataene som skal lagres.

Sikkerhetstjenester

En arkitekturstrategi som innebærer migrering av infrastruktur, tjenester og databaser til sky, er i veldig stor grad avhengig av robuste og velfungerende sikkerhetstjenester. For Enova er en robust sikkerhetsarkitektur helt avgjørende for realisering av gjeldende målarkitektur.

Av komponentene diskutert her har vi gjort noen valg og satt sammen en målarkitektur vi tror vil gi Enova nødvendig fleksibilitet og kapasitet til å løse fremtidens behov.



Hva ble testet?

For å gjennomføre denne PoC'en tok vi ut prøvelisenser på Microsoft Dynamics 365 Customer Engagement (tidligere CRM/XRM). Dette ga oss tilgang til applikasjonsutviklingsverktøyene som kommer som en del av Dynamics, men også en informasjonsmodell (CDM) og nødvendig funksjonalitet for å betjene denne (PowerApps).

Merk at denne PoC'en kun bekrefter deler av målarkitekturen og ikke omfatter testing av bl.a. «hendelsesbuss» og funksjonalitet for en «API Gateway».

Utvidelse av en standardentitet

En felles informasjonsmodell fra en leverandør kan ikke passe alle. Et svært sentralt punkt er derfor muligheten man har til å utvide en standard entitet med egne attributter. Vi testet dette konseptet med å mappe kundeobjektet fra Enovatilskuddet (løsning for privatmarkedet) inn i kundeobjektet i CDS. CDS tilbyr funksjonalitet for å gjøre dette direkte fra web-grensesnitt, enten via Dynamics eller direkte i PowerApp. Grensesnittet oppleves som enkelt og disse oppgavene kan fint utføres av Enova selv (ingen «koding» nødvendig).

Etablering av egen entitet

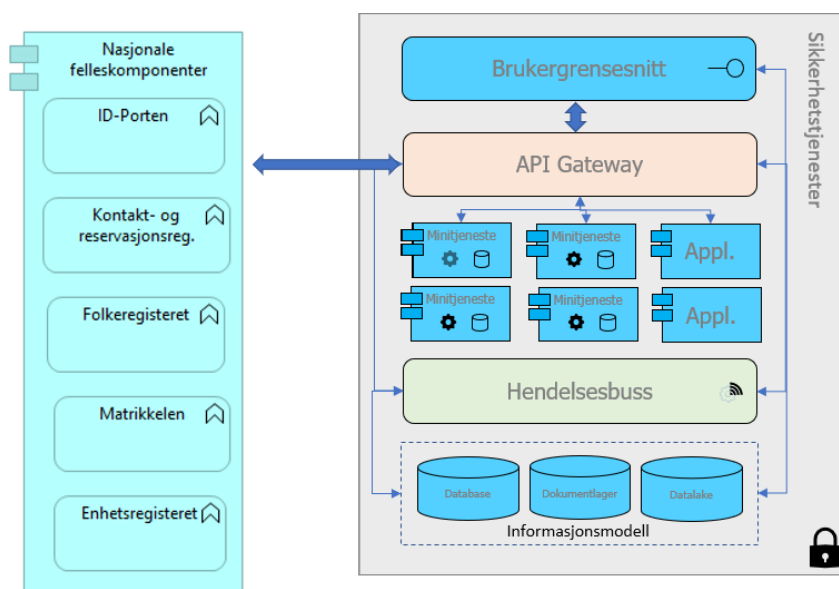
I tillegg til å tilby muligheter for å legge til attributter i standardentiteter (i CDM) gir løsningen mulighet til å legge til (utvide) modellen med egne entiteter. Dette anses også som en sentral funksjonalitet for Enova. Vi eksperimenterte en god del knyttet til dette punktet og konsentrerte oss om entitetene «Markedsendringsmål» og «Sektor». På samme måte som for punktet over oppleves grensesnittet som enkelt og dette er fint noe Enova selv kan vedlikeholde. Felles for begge punktene at dette fordrer en tydelig og godt dokumentert informasjonsmodell i forkant (før man begynner å etablere entiteter og attributter). Det er også mulig å etablere en entitet basert på strukturen i de dataene man ønsker å importere, løsningen vil da lage en entitet med attributter som matcher de data du ønsker å importere (utføres vha Power Query).

Test av grensesnitt mot CDM

Et viktig moment i CDM er at man så enkelt som mulig kan opprette, endre og slette innhold i tabellene som inneholder alle Enovas data. Dette er det tatt høyde for i Microsoft sin løsning. Bl.a. har alle entiteter og attributter

i datamodellen et ferdig REST Api «ut av boksen», dette grensesnittet ivaretar også sikkerhetsmodellen som settes opp (ingen uautorisert tilgang til data).

Sammen med muligheten til å etablere micro-/minitjenester og funksjoner kan dette grensesnittet for eksempel benyttes til å lage felles tjenester/funksjoner som kan benyttes mot eksterne tjenester. Dagens integrasjon mot nasjonale felleskomponenter (ID-porten, folkeregisteret, matrikkelen med flere) kan med denne arkitekturen lages som en samling tjenester som alle moduler og applikasjoner kan benytte seg av, istedenfor at hver applikasjon lager en integrasjon hver. Se figur under.



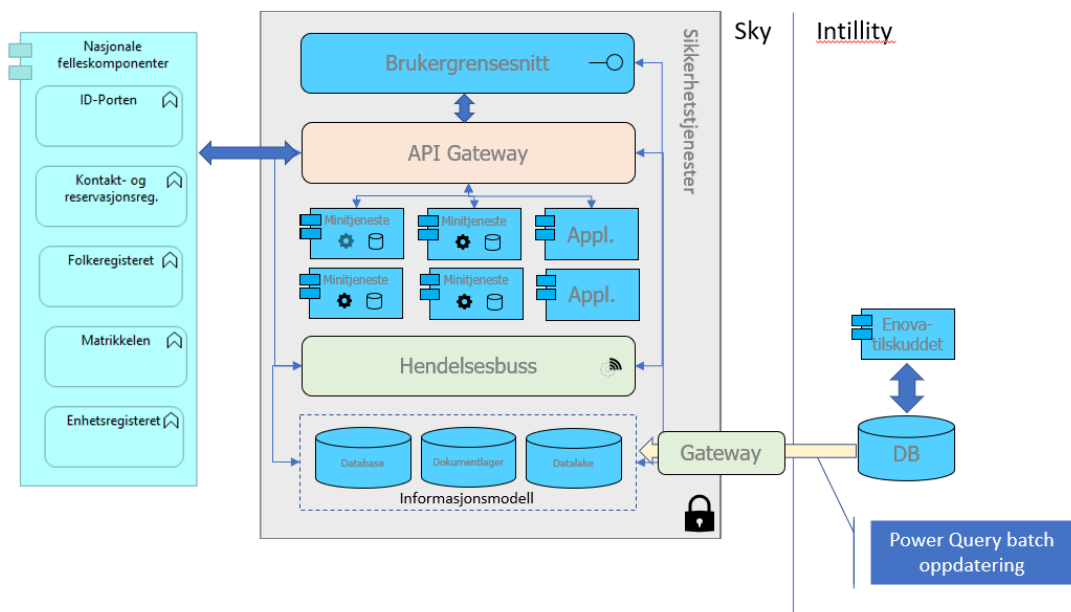
Eksempler på tjenester:

- Autentiser søker
- Oppdater adresse
- Hent matrikkeldata
- Sjekk org-nummer

I tillegg til REST Api benytter denne løsningen Power Query (M-language) som «spørrespråk» mot de kildene man trenger å hente data fra. Løsningen har allerede ferdige Connectorer mot en hel rekke kilder og via Enovas PowerBI Gateway kan vi også koble oss til interne databaser for eksport av data. Dette er samme teknologi som benyttes av PowerBI.

Ferdig oppsatte integrasjoner (Power Query) kan planlegges og kjøres ved faste intervall («batch» kjøring). Løsningen har også funksjonalitet for «delta-last», hvor bare nye og oppdaterte rader importeres slik at man unngår en full synkronisering hver gang. Dette ble testet i praksis ved å periodisk overføre nye kunder fra Enovatilskuddet til kundetabellen i CDM. Importerte kunder dukker opp i Dynamics og kan jobbes med (berikes) med standardfunksjonaliteten der.

Skissen under viser testoppsettet vi benyttet for å laste kunder fra Enovatilskuddet ved hjelp av spørrespråket Power Query. Det er «Gateway» komponenten som autentiserer bruker inn til våre interne systemer. Merk at denne integrasjonen kjører med en gitt frekvens og ikke oppdaterer i det øyeblikk en ny kunde opprettes.



Videre har vi testet import av data fra Excel-filer, Sharepoint Online lister og egendefinerte PowerApps. Med unntak av noen mindre tekniske utfordringer fungerte dette som forventet.

Når det gjelder integrasjon av 3. parts hyllewareprodukter (Websak, Maconomy, Basware, Visma, Limepro osv) tilbyr plattformen flere måter å realisere dette på. Hvordan dette utføres (hvilken metode som benyttes) avhenger i stor grad av hvilke grensesnitt som tilbys av 3. part applikasjonen og hvor «tidskritiske» dataoppdateringer er. Moderne løsninger tilbyr ofte grensesnitt (REST Api) som er enkle å integrere seg mot (microtjenester direkte mot 3. part), mens for andre må vi kanskje gå rett på databasen eller til og med basere oss på import/eksport av flate filer. Vi fikk ikke testet akkurat denne problemstillingen som en del av PoCen, men en av de to variantene vi har omtalt her kan benyttes til å integrere seg mot 3. part løsninger.

Vi testet også PowerBIs egne «connectorer» mot CDS, disse er i foreløpige utgaver men vi ser likevel et betydelig potensial her. En komplett og utvidet CDS vil forenkle rapportering betydelig.

Foreløpig konklusjon

CDM/CDS versjon 2.0 (som vi testet) ble gjort tilgjengelig i april 2018. Det er med andre ord fortsatt en umoden løsning, noe løsningen også bærer litt preg av. Vi savner bl.a. et bedre og mer effektivt verktøy for konfigurering, samt at en del av skjermbildene mangler noe funksjonalitet og i enkelte tilfeller også feiler.

På tross av dette mener vi at CDM/CDS med tilhørende funksjonalitet fra Azure (mikro-/minitjenester, funksjoner og tjenestebuss) er et godt eksempel på hvordan Enovas målarkitektur kan realiseres.

Alternativ til Microsoft Dynamics 365

Arbeidet i PoCen har vist oss at vår målarkitektur både er realiserbar og relevant for fremtiden. Parallelt med at vi har testet noen av Microsoft sine løsninger har vi også undersøkt hvordan Oracle nå tilbyr applikasjoner og tjenester for å flytte sine kunder over på skyløsninger. Men unntak av litt andre begrep og gruppering av tjenester er grunnprinsippene relativt like (som beskrevet over). Oracle bekreftet i møte at vår arkitektur er realiserbar på deres plattform.

Det er grunn til å tro at også andre store skyleverandører, som for eksempel Google, Amazon og IBM, er i stand til å levere infrastruktur og tjenester som tilfredsstillende Enovas målarkitektur. Dette har vi kun undersøkt ved å lese litt

på nettsidene deres, og finner bl.a. at IBM sin Microservices Architecture ser ut til å ha samme oppbygging, ref. https://www.ibm.com/cloud/garage/files/Microservices_Architecture_Overview.pdf

Kostnads- og risikovurdering

Kostnadmessige konsekvenser ved skytjenester er ikke vurdert i detalj, men både egen erfaring og trender i markedet tilsier at en IT-plattform i sky er mer kostnadseffektivt enn alternativet. I tillegg til lavere tjenestekostnader (produktkostnader) vil Enova kunne benytte komponenter som allerede er i bruk (og testet) av veldig mange andre kunder, og dermed unngå kostnader forbundet med egen konfigurering, test og forvaltning.

Prismodellene til de ulike skyleverandørene er i tillegg veldig komplekse og under stadig endring, det er derfor krevende å på nåværende tidspunkt finne ut hva en realisering av valgt målarkitektur vil medføre av kostnader. Bruk av OpenSource produkter er et reelt alternativ for å realisere deler av behovene i ny IT-arkitektur.

I tillegg til at skytjenester sannsynligvis er billigere enn alternativet, vil en oppdeling av arkitekturen (fra «silopplikasjoner» til minitjenester) også bidra positivt på totalkostnadene ved denne målarkitekturen. Bl.a. fordi Enova i større grad kan gjøre småskrittsutvikling der det er behov uten å endre og regresjonsteste store deler av IT-plattformen og på den måten redusere det vi kan kalle for «teknisk gjeld» knyttet til utdatert teknologi (både maskinpark og programvare).

Full klarhet i kostnadsbildet knyttet til ny arkitektur basert på skyleverandør er selvfølgelig viktig å holde under kontroll, men samtidig krevende å ha full oversikt over på nåværende tidspunkt. Dette blir gradvis tydeligere etter hvert som vi kommer i dialog med markedet og vi anbefaler derfor en konsekvensutredning knyttet til dette på et senere tidspunkt.

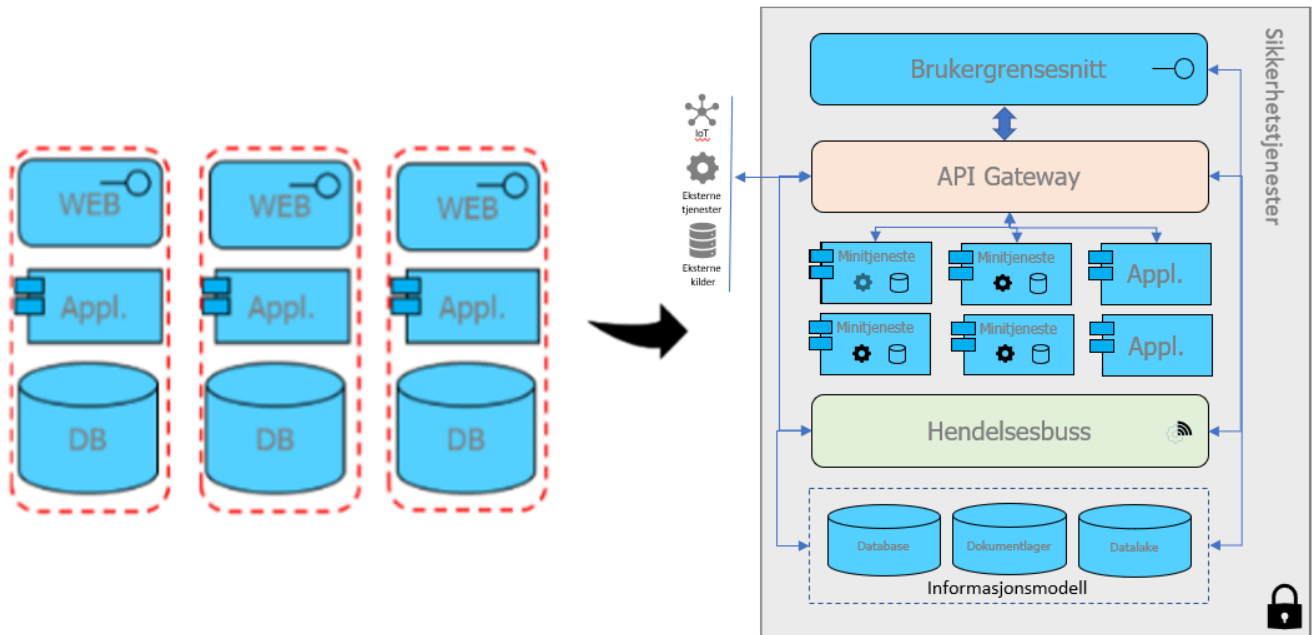
Migrering av IT-tjenester til skyløsninger introduserer nye risikoer som man ikke er utsatt for når man selv eier benyttet infrastruktur. Mindre kontroll over egne data, økt kompleksitet og redusert innsikt og kontroll er alle eksempler på risikoer som i større grad er tilstede i rene skyløsninger.

I tillegg er det, i forbindelse med migrering til sky, viktig å huske at skyleverandører benytter en delt modell for sikkerhet i skyen. Leverandørene aksepterer ansvar for noe av sikkerheten, mens noe blir et delt ansvar mellom kunde og leverandør. Likevel er det noen aspekter knyttet til sikkerhet som utelukkende er kundens ansvar. Effektiv og god sikkerhet i skyen avhenger av innsikt og forståelse for hvordan sikkerheten skal ivaretas. En kundes manglende evne til å møte deres ansvar knyttet til sikkerhet i skyen, er den vanligste årsaken til sikkerhetsbrudd i skybaserte systemer.

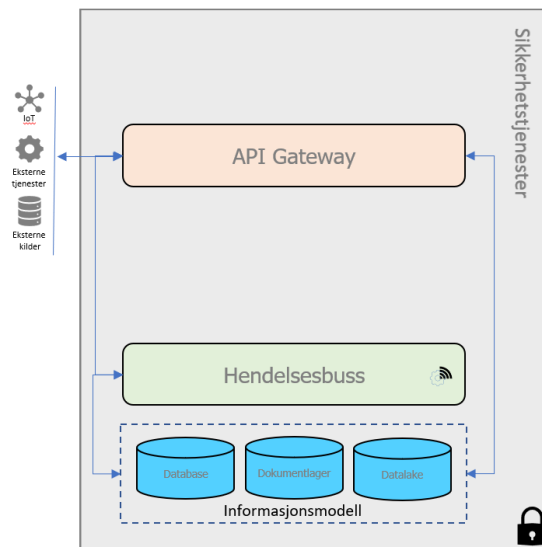
For Enova er derfor viktig å ha et kontinuerlig fokus på risikoen knyttet til realisering av nytt mål bilde for IT-arkitekturen, dette vil vi blant annet sikre ved å gjennomføre og benytte Risiko – og sårbarhetsanalyser (ROS-analyser) som verktøy for å minimere den risikoen vi blir utsatt for.

Veien videre

Spørsmålet blir da: Hvordan skal Enova gå frem for å realisere ønsket målarkitektur?



Vår hypotese er at det første steget mot å realisere vårt mål bilde er å etablere det vi kan kalle grunnmuren i målarkitekturen. I denne sammenheng tror vi det handler om å etablere en hendelsesbuss, en API gateway og nødvendige løsninger for datalagring og sikkerhet i skyen.



Dette kan enten innføres/etableres som en egen anskaffelse eller sammen med at vi enten innfører en ny standardapplikasjon eller en sentral minitjeneste. Deretter handler det om å bygge stein for stein og samtidig sørge for at alle nye endringer som etableres passer inn i vårt mål bilde, samt legge en plan for alle eksisterende løsninger slik at de (om mulig) passer inn i og unnytter grunnmuren på best mulig måte.



Med begrensede egne IT-ressurser er det også avgjørende at Enova etablerer et samarbeid med en eller flere strategiske partnere som kan bistå, rådgi og konfigurere den tekniske løsningen.

Suksessfaktorer

- Valg av riktig partner blir avgjørende
- Kontroll på risikobildet i forbindelse med migrering til sky
- God forankring internt
- Langsiktighet i plan og gjennomføring

Konsekvens

- Viktigheten av en svært god informasjonsmodell øker kraftig
- Streng kontroll på hvem og hvordan man endrer «kjernen», altså informasjonsmodellen
- Man får en «ramme» av funksjoner løsningen som man må forholde seg til; høyere grad av gjenbruk
- Når plattformen kommer med standarder som kan erstatte egne løsninger, skal disse fortrinnsvis innføres

Konklusjon og anbefalinger

Vårt arbeid med denne «Proof of concept» har gitt oss flere bekreftelser. Først og fremst har vi fått en god verifisering av at flere store skytjenesteleverandører bygger opp sine tilbud til kunne møte behovene vi har skissert i vår målarkitektur. Et langsiktig mål om å etablere en IT-arkitektur, slik den er skissert i vår målarkitektur, er riktig vei å gå for Enova.

På nåværende tidspunkt er det krevende å ha full oversikt over totalkostnadene knyttet til ny etablering av ny målarkitektur basert på skykomponenter, vi er derimot relativt sikre på at dette er mer kostnadseffektivt enn alternativet. Det er videre ingen tvil om at en migrering av alle data og IT-tjenester til sky utsetter oss for en økt risiko og medfører en økt kompleksitet i vår totale IT-infrastruktur. Vi anbefaler derfor at det iverksettes en egen ROS-analyse tidlig i prosessen for å redusere og kontrollere risiko og kompleksitet forbundet med en «sky-migrering» og at det gjennomføres en kostnadsvurdering på et senere tidspunkt.

Økt kompleksitet og økt tilgjengelighet på nye funksjoner og tjenester (AI, robotisering med videre), som sannsynligvis har høy forretningsverdi, understreker også behovet for å styrke forankring og målrettet arbeid med den totale forretningsarkitekturen i Enova. Flytting av all IT-infrastruktur til sky vil påvirke alle prosjekter i Enova som har behov for IT-tjenester.

Enova er godt i gang med sitt prosjekt for helhetlig virksomhetsstyring og organisasjonsutvikling. Sammen med en ambisjon om å bidra til den nødvendige samfunnsstransformasjonen mot et lavutslippssamfunn, vil dette medføre betydelige løfter for Enova på mange måter, også i «maskinrommet». Vi anbefaler at foreslått målarkitektur, funnene i denne studien og virksomhetsarkitektur generelt legges til grunn for framtidige anskaffelser av, og senere på, en digital forretningsplattform. Enova kan dermed utnytte kraften i dette initiativet til å ta det første og viktige skrittet på vei mot å realisere vårt målbilde for en moderne, fleksibel og fremtidsrettet IT-arkitektur.

- Målarkitekturen er realiserbar og riktig vei å gå for Enova
- En økonomisk konsekvensutredning må gjøres
- Bruk av skyløsninger øker risiko og krever en ROS-analyse for å håndtere risiko
- Målarkitekturen påvirker alle framtidige IT-prosjekter
- Kompleksiteten øker; krever økt kunnskap, god styring og krever sterk forankring - fra hele Enova
- Rapporten bør benyttes som underlag til anskaffelsen av digital forretningsplattform

Ordliste

Begrep/forkortelse	Betydning
Målarkitektur	Uttrykker et målbilde og en masterplan for en fremtidig IT-arkitektur. Målarkitekturen er forankret i forretningens målsetninger og visjon. En målarkitektur benyttes bl.a. for å sikre at nye IT-prosjekter «passer inn» og understøtter ønsket forretningsutvikling.
Entitet	En entitet er et tydelig definert og lett identifiserbart objekt (konkret eller abstrakt) som man ønsker å lagre informasjon om. «Kunde», «ordre», «saksbehandler», «søknad» er eksempler på entiteter.
CDM	«Common data model» - Produktnavn på Microsofts felles informasjonsmodell.
CDS	«Common data services» - Produktnavn på Microsofts tjenester for å implementere en CDM.
Dynamics 365	Overbyggende produktnavn på alle «applikasjoner» som Microsoft leverer til sine kunder. Vil på sikt inkludere alle tradisjonelle applikasjoner som HR, salg, service, ERP osv.
Azure	Microsoft sin sky løsning.
PoC	Proof of Concept er en realisering av en bestemt metode eller ide for å demonstrere dens mulighet, eller en demonstrasjon i prinsippet med sikte på å verifisere at et konsept eller en teori har praktisk potensiale.
PowerApp	Produktnavn fra Microsoft. Verktøy laget for sluttbrukere som gjør det relativt enkelt å lage enkle forretningsapplikasjoner.
SOA	«Service Oriented Architecture» - Et samlebegrep for tjenesteorientert arkitektur. (Populært buzz-word for 10 år siden i bransjen. Red. anm.)
REST Api	Svært enkel protokoll for å tillate kommunikasjon mellom web-baserte applikasjoner. Svært mye brukt i dag. REST står for «Representational State Transfer» og API står for «Application Programming Interface»
Power Query	Spørrespråk mye brukt av Microsoft.
On-premise («On-prem»)	Ofte benyttet begrep (også på norsk) for å beskrive tradisjonelle IT «maskinrom» som er knyttet til egen eller leid maskinpark.

Vedlegg 1: Utfyllende om risiko ved skyløsninger

Her er eksempler på noen spesifikke risikoer fra en artikkel(blogg) publisert av Timothy Morrow ved Carnegie Mellon University. I dette kapittelet (fritt oversatt og forkortet) tar vi kun med noen utvalgte risikoer, hele artikkelen kan leses [her](#).

#1 Kunder har redusert innsikt og kontroll

Når organisasjoner flytter innhold og infrastruktur til sky mister man noe oversikt og kontroll over egne ressurser. Ved bruk av eksterne skytjenester flyttes noe av ansvaret for å ivareta policyer og infrastruktur til skytjenesteleverandøren, og kunden mister muligheten til å benytte nettverksbaserte verktøy for logging og overvåkning.

#2 «On-demand» selvbetjeningsløsninger forenkler uautorisert bruk internt

Skytjenesteleverandører gjør det veldig enkelt å aktivere nye tjenester. Dersom man ikke etablerer gode rutiner knyttet til dette kan man risikere at brukere selv aktiverer nye tjenester uten godkjenning fra IT.

#3 Kompromittering av internettbaserte grensesnitt for administrering

Det faktum at en skyløsning alltid er tilgjengelig på internett, betyr også at alle grensesnitt for administrering av alt innhold også er tilgjengelige på internett. Hvis disse tilgangene kan hackes eller på andre måter utnyttes medfører dette en betydelig trussel for virksomheten. Med «on-premise» IT har vi ikke denne risikoen, bl.a. fordi man har dedikerte og sikre brannmurer.

#5 Datasletting blir ufullstendig

Kunden mister oversikt hvor egne data fysisk er lagret. I en skyløsning kan data spres over flere forskjellige lagringsmedier på mange forskjellige lokasjoner. Det kan derfor være vanskeligere å forsikre seg om at dataene faktisk er slettet.

#7 «Vendor Lock-in» vanskeliggjør flytting av tjenester fra en leverandør til en annen

Selv om grunnprinsippene for arkitekturen er relativt like for alle skyleverandørene, er selve implementeringen (koden) ulik. Med utstrakt bruk av funksjonalitet, tjenester eller grensesnitt blir et eventuelt skifte til en ny tjenesteleverandør en veldig krevende oppgave.

#8 Økt kompleksitet stiller høyere krav til IT

En migrering til sky kan introdusere økt kompleksitet i IT forvaltningen.

Oppsummering

I forbindelse med migrering til sky er det viktig å huske at skyleverandører benytter en delt modell for sikkerhet i skyen. Leverandørene aksepterer ansvar for noe av sikkerheten, mens noe blir et delt ansvar mellom kunde og leverandør. Likevel er det noen aspekter knyttet til sikkerhet som utelukkende er kundens ansvar. Effektiv og god sikkerhet i skyen avhenger av innsikt og forståelse for hvordan sikkerheten skal ivaretas. En kundes manglende evne til å møte deres ansvar knyttet til sikkerhet i skyen, er den vanligste årsaken til sikkerhetsbrudd i skybaserte systemer.