

Beste Praksis – Arkitektur for hendelser i felles økosystem

Skrevet av: Helle Stedøy og Morten Græsby / Digitaliseringsdirektoratet/Felles Økosystem

Logg:

Dokument opprettet	Disposisjon og skrivestart	Dato
Versjon 0.7	Utkast sendt til ASR	29.11.2022
0.71	Lagt inn kilder og justert	22.12.2022
0.8	Introduksjonstekst. Klar til utsending	6. 02.2023

Innhold

Introduksjon til arbeid med «Beste praksis – arkitektur for hendelser i felles økosystem»	2
Høring.....	2
Videre arbeid med «Beste praksis – Arkitektur for hendelser i felles økosystem»	2
1 Innledning.....	3
1.1 Utfordringsbildet	3
1.2 Test	3
1.3 Hvorfor er det smart å være hendelsesdrevet	5
1.4 Behov.....	6
1.5 Knytning til de overordnede arkitekturprinsippene	8
2 Arkitektur for hendelser i Felles Økosystem	9
2.1 «Hendelsesdrevet arkitektur» i en verden som består av hendelser	9
2.2 «Hendelsesdrevet arkitektur – logisk»	10
2.3 Hendelsesmønstre i «Arkitektur for hendelser i Felles Økosystem».....	11
3 Samspill i Felles Økosystem	17
3.1 Teknisk Interoperabilitet	17
3.2 Semantisk Interoperabilitet.....	18
4 Vedlegg: Eksempel på behov og scenario	19
4.1 Dødsfall og Arv.....	19
4.2 Fremtidens innkreving.....	19
4.3 Kunngjøringer fra Foretaksregisteret, Regnskapsregisteret, Partiregisteret, Konkursregisteret og Stiftelsestilsynet.....	20
4.4 Folkeregisteret og Innbyggerkontakt	20
4.5 Byggesøknad.....	21
4.6 E-Bevis	21
4.7 Advokatregisteret og representasjon	21
5 Vedlegg – Leseliste og kilder	22

Introduksjon til arbeid med «Beste praksis – arkitektur for hendelser i felles økosystem»

Arbeidet med en «beste praksis» ble igangsatt av Digdir ved Morten Græsby og Helle Stedøy høst 2022, med formål om at arkitektur for hendelser kan bidra til å oppnå mål om økt grad av samhandlingsevne i felles økosystem.

Vi ønsker at resultatet av dette arbeidet blir tatt i bruk i fremtidig tjenesteutvikling fra tidlig konseptfase til endelige løsninger. Vi valgte derfor at utgangspunktet for en beste praksis skulle være overordnet fra behov og konsept og ned til det mer løsningsnære, slik at innholdet kan brukes og forstås av alle som arbeider med tjenesteutvikling.

Arbeidet har tatt utgangspunkt i noen kjente behov og reelle forretningscase.

Dokumentet begynner derfor å innlede om bakgrunn og utfordringsbildet, hvilke behov som må dekkes og knytning til felles økosystem. Deretter går vi inn på mer løsningsnære mønstre. Vi har valgt å ta med noen få mer tekniske eksempler, for å konkretisere innholdet.

Arkitektur- og Standardiseringsrådet har vært referansegruppe i arbeidet, og har også gitt tilbakemeldinger på innholdet i forkant av denne høringen.

Høring

Vi ber om at hørings svar sendes innen 24. Mars. Svar gjøres i følgende skjema:

<https://forms.office.com/e/qnqxUsYX9L>

Videre arbeid med «Beste praksis – Arkitektur for hendelser i felles økosystem»

Merk at dette er en ikke-ferdig versjon («versjon 0.8») som vi ber om hørings svar på. Når vi har fått bearbeidet innspill vil en versjon 1.0 publiseres på Digdir sine nettsider som Web-sider, ikke som et statisk dokument.

Vi ønsker videre en smidig tilnærming til temaet, slik at nye behov og tema kan detaljeres ytterligere innenfor rammen hendelsesdrevet arkitektur, som f.eks.:

- Tilgangsstyring av hendelser
- IoT
- Komplekse hendelser (hendelser basert på analyse, monitorering av data og/eller flere enkle hendelser i sammenheng)
- ...

1 Innledning

Økosystem for nasjonal digital samhandling og tjenesteutvikling, skapes gjennom måten vi samhandler på, på tvers av sektorer, fagområder og bransjer.

For å utvikle sammenhengende tjenester i økosystemet er det helt sentralt at vi har evnen til å ta brukerperspektivet, og at vi etablerer og sikrer samarbeid og styring mellom aktørene – Vi starter sammen, vi former sammen og vi leverer sammen!

Samtidig må vi erkjenne at aktørene som samhandler i økosystemet i stor grad er autonome og at tjenester og data er distribuert. For å levere sammenhengende tjenester til innbyggere og virksomheter på en dynamisk og effektiv måte, må vi etablere evnen til å fange opp når en hendelse inntreffer. Vi må kunne publisere hendelser og gjør det mulig for andre aktører å abonnere på disse hendelsene. Tjenesteeierne må kunne tilgjengeliggjøre beskrivelser av sine tjenester slik at tjenestene kan sammenkobles eller integreres med andre tjenester og danne tjenestekjeder. Sammenkoblingene bør skje på en måte som gir størst mulig fleksibilitet for de involverte aktørene.

1.1 Utfordringsbildet

Hovedutfordringer

1. Sammenhengende tjenester utvikles i (større) siloer, og det er risiko for at disse blir lite fleksible med hensyn til videreutvikling og nye kontinuerlige behov.
2. Manglende kunnskap og kompetanse på hva, hvorfor og hvordan «hendelser» må håndteres fra konsept til løsning, og hvilken effekt det vil ha på fremtidig tjenesteutvikling
3. Det finnes ingen dominerende tilnærming til utveksling av hendelser, meldingsformat eller meldingskommunikasjon på tvers av hendelsesdrevet integrasjonsteknologi. Å forstå hvordan aktørene skal utveksle hendelser er nøkkelen til å vurdere fordelene vi kan oppnå.

1.1.1 Sammenhengende tjenester utvikles i større siloer

Sammenhengende tjenester utvikles i større siloer der et begrenset antall aktører samarbeider om å utvikle en sammenhengende tjeneste. Sammenhengende tjenester blir i stor grad orkestrert. Dette fører til redusert mulighet for gjenbruk, og endringer i den sammenhengende tjenesten krever mye koordinering fra de involverte aktørene.

1.2 Test

Eksempler på sammenhengende tjenester som orkestreres ([se vedlegg](#)):

- Byggesøknad
- Oppgjør etter dødsfall
- eBevis

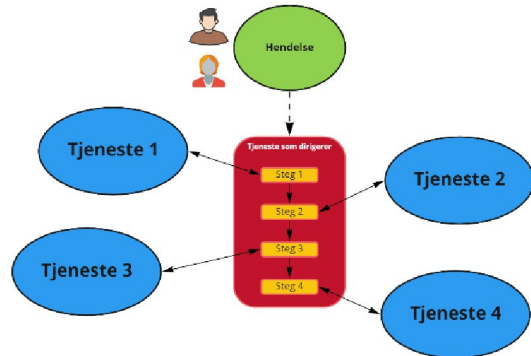
1.2.1.1 Orkestrering av tjenester

Ved orkestrering er det en sentral prosess (dirigent/koordinator) som styrer interaksjonene med tjenestene som inngår i tjenestekjeden. Den sentrale prosessen bestemmer spillereglene for deltakelse i kjeden. Den sentrale prosessen definerer kjeden fra begynnelse til slutt.

Det er sterk avhengighet mellom tjenestene.

Integrasjonene mellom tjenestene gir lite fleksibilitet, skalerer dårlig og det er krevende å få til samhandlingsevne mellom mange tjenester.

En endring i en tjeneste i kjeden vil ofte føre til behov for endringer i de andre tjenestene.



1.2.2 Manglende kunnskap og kompetanse

Det er manglende kunnskap om hva en hendelse er. Mange blander sammen aktiviteter, kommandoer og hendelser. I tillegg blir livssituasjoner blir noe misvisende kalt livshendelser. En hendelse er en følge av en planlagt eller uplanlagt aktivitet. Aktiviteten kan utføres av en sluttbruker, en saksbehandler eller i en automatisert prosess. Kompetanse om hvordan vi kan utnytte hendelser i teknologien er mangelvare, og i et teknisk perspektiv vil en hendelse føre til behov for tjenester som bruker data, oppretter eller endrer data – slik oppstår en ny hendelse. En hendelse er noe som har skjedd for eksempel

- Barnet er blitt syk
- Vedtak om opphold fattet
- Nabovarsel er sendt
- Enkeltmannsforetak er registrert i foretaksregisteret

En kommando er styrende og formidler noe som skal skje, ved en kommando forventes en respons eller en kvittering tilbake.



- En hendelse er en endring i en tilstand (noe som har skjedd og beskrives i fortid)
- Tilbyder har fullt eierskap til hendelsen
- Det er kun én tilbyder av en hendelse
- Det er ingen eller mange konsumenter av en hendelse



- En kommando er en forespørsel som krever en handling og en respons
- Mottaker tar eierskap til å utføre handlingen som kommandoen krever og gi respons
- Det er kun én mottaker som kan respondere på en forespørsel
- Det kan være flere produsenter av forespørsler til samme mottaker

1.2.3 Ingen dominerende tilnærming

I dag finnes det ingen felles metode eller beste praksis for utveksling av hendelser på tvers av virksomheter og sektorer. Det er fare for at sektorer og virksomheter løser dette på hver sin måte, og at samhandling på tvers ved hjelp av hendelser blir komplekst og i verste fall umulig.

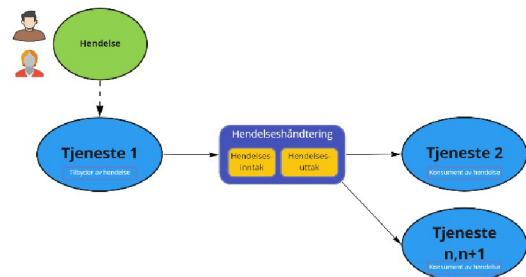
Utteksling av hendelser som samhandlingsform kan bidra til en rekke fordeler for aktørene. Det gir økt fleksibilitet og modularitet i samhandlingen og det vil bidra til svært løse koblinger mellom tjenestene som leveres av de ulike aktørene.

1.3 Hvorfor er det smart å være hendelsesdrevet

For å lykkes med sammenhengende tjenester må tjenestene kunne utvikles mer dynamisk og uavhengig av hverandre og baseres på løse koblinger og koreografi.

1.3.1.1 Koreografi av tjenester

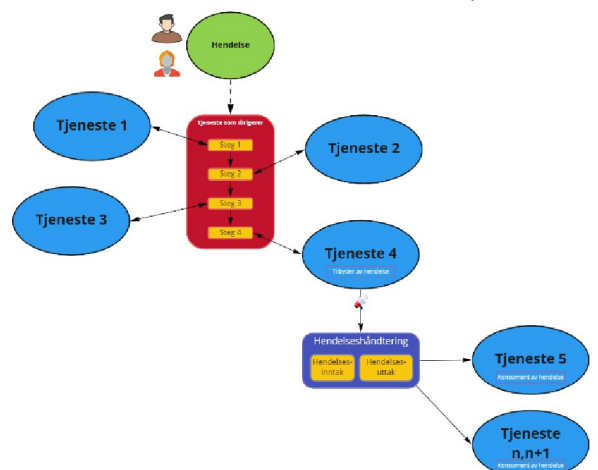
Ved koreografi er det ingen sentral eller helhetlig prosess som styrer interaksjonene med tjenestene som inngår i tjenestekjeden. Hver tjeneste i kjeden trigges av et resultat, eller en hendelse fra en eller flere andre tjenester. Dermed har den enkelte tjeneste kontroll på sin egen integrasjon med de andre tjenestene. Ved koreografi baseres arkitekturen på løse koblinger som gir økt fleksibilitet. Tjenester som inngår i kjeden, kan enklere byttes ut, og de kan inngå eller gjenbrukes i flere ulike tjenestekjeder. De enkelte tjenestene trenger ikke å vite om hverandre. Ved koreografi reduseres kompleksitet og sårbarhet.



En beste praksis for «Arkitektur for hendelser i felles økosystem» vil bidra til at virksomhetene kan utveksle og agere på hendelser på tvers. For offentlig sektor er dette en relativt ny måte å samhandle på, på tvers av virksomheter og sektorer. Ved å basere arkitekturen på hendelser vil tjenester kunne tilbys selvstendig eller i en tjenestekjede og nye behov vil kunne støttes i tilnærmet sanntid og uten endringer i tilstøtende tjenester. Tjenester og informasjon vil kunne tilpasses til brukerens behov smidig og kontinuerlig, og det vil være mulig å agere proaktivt overfor bruker.

1.3.1.2 Orkestrering og koreografi av sammenhengende tjenester

For sammenhengende tjenester som skal følge en fastlagt prosess der det er viktig å ha kontroll på alle involverte aktører og deres bidrag inn i den sammenhengende tjenesten, vil orkestrering være det beste alternativet. Det vil allikevel være viktig å ikke orkestrere mer enn det som er absolutt nødvendig. Vurder derfor om en kombinasjon av orkestrering og koreografi kan være den beste tilnærmingen for å levere den sammenhengende tjenesten.



1.3.2 Hypoteser

Arkitektur for hendelser vil bidra til en mer effektiv og enklere utvikling og forvaltning av tjenester og vil kunne gi store kostnadsbesparelser. Det vil være mindre punkt-til-punkt integrasjon, svært løse koplinger og færre avhengigheter mellom tjenestene som tilbys. Dette vil kunne bidra til bedre ytelse i tjenesteleveransene og være et viktig bidrag for økt dataminimering. Det vil være behov for økt kompetanse om hendelsesdrevet arkitektur og etter hvert som modenheten øker vil det være et økende behov å kunne håndtere hendelser i felles økosystem.

Ved hjelp av en hendelsesdrevet arkitektur vil innovasjonskraften øke, både i offentlig-, kommunal- og privat sektor. I dag baseres store deler av samhandlingen på tvers av aktører, på kommandoer og API-oppslag. Tilbyder og konsument må kjenne til hverandre og hvilke behov den andre part har. Ved å basere samhandlingen på hendelser trenger ikke tilbyder å kjenne til konsumentene, hverken hvem de er, hvor mange de er eller hvilke behov de har. Konsumentene kan abonnere på hendelser som er interessant for dem og benytte disse til å skape nye og innovative tjenester. Det ligger et enormt potensial i dette. For eksempel kan et adressebytte være interessant for svært mange ulike aktører, både statlige, kommunale og private tjenester. En kommune kan abonnere på «nye innbyggere» og tilby informasjon og tjenester som er aktuelle for de nye innbyggerne.

1.4 Behov

Vi lever i en tid der store teknologigiganter og servicenæringen tilbyr proaktive og brukertilpassede digitale tjenester. Innbyggere og virksomheter forventer tilsvarende fra det offentlige.

Innbyggerne forventer at hendelser og endringer i livet proaktivt skal påvirker hvilke tjenester som tilbys, og at tjenestene er persontilpassede og relevante. Innbyggerne vil ha sammenhengende tjenester, de ønsker å slippe å forholde seg til mange forskjellige aktører og nettsted. Informasjon må tilbys helhetlig og enhetlig ut ifra en gitt kontekst.

Virksomheter ønsker å tilby tjenester og informasjon i alle de sammenhenger som er relevante for en bruker. Dette skal kunne skje uten endring av de løsningene eller tjenestene brukeren er i, også når disse tjenestene er utviklet av andre aktører. Dette innebærer at relevante tjenester eller informasjon, utviklet av andre, kan tilbys som en del av egne tjenester. Virksomheter har behov for å finne og få oversikt over tjenester som tilbys av andre aktører, og de må kunne få oversikt over hvilke tjenesterresultat eller hendelser som kan trigge egne tjenester.

Det er et behov for en beste praksis for utveksling av hendelser på tvers av løsninger og sektorer, og der er behov for løsning(er) for å håndtere og utveksle hendelser, både i sektorene selv og som en fellesløsning.

Behovene for å utveksle hendelser er i hovedsak knyttet til to generelle scenario:

1. Tjenestekjeder - hendelser som oppstår etter at en tjeneste er utført, der en eller flere andre tjenester kan/skal bruke resultatet fra denne.
2. Tilgang til hendelsesdata fra datakilder, der det ikke nødvendigvis er en veldig klar tjenestekjede (for eksempel fra registre eller datakilder som tilbyr sammenstilt informasjon)

I tillegg kommer hendelser basert på telemetri og monitorering, og såkalte «komplekse hendelser». Dette vil vi ta inn i videreutviklingen av en «beste praksis»:

- Hendelser basert på telemetri og monitorering knyttes ofte til «IoT – Internet of things»

- «Komplekse hendelser»: hendelser basert på analyse og monitorering av data og/eller flere enkelthendelser

1.4.1 Behov for å koble sammen tjenester i sanntid?

For å kunne realisere dynamiske tjenestekjeder som utveksler informasjon mellom distribuerte løsninger på tvers av virksomheter uten komplekse og tette integrasjoner, må tjenestene kunne kobles sammen i tilnærmet sanntid. Et tjenesteresultat eller en hendelse må dermed «pushes» fra en tilbyder-tjeneste til en konsument-tjeneste i det øyeblikket endringen/hendelsen har skjedd.

Eksempler på behov for hendelser i sanntid (scenario 1):

- Advokatregisteret
 - **Endring/hendelse:** «Advokat mistet bevilgning»
 - **Behov:** Konsumenter må agere raskt! Varsel/Melding om hendelsen må sendes ut i det endringen oppstår. Det er behov for en «Push» av hendelsen til konsument.
- E-bevis
 - **Endring/hendelse:** Leverandør må samtykke til oppdragsgiver om innsyn i egne skatteopplysninger
 - **Behov:** For raskere behandling er det behov for at oppdragsgiver får tilgang til skatteopplysninger umiddelbart. Det er behov for en «Push» av hendelsen til konsument.

I noen tilfeller er det mer hensiktsmessig at konsumenten selv kan velge når den ønsker å lese hendelsen og behovet for å få tilgang til en hendelse i sanntid er ikke til stede.

Eksempler på behov for hendelser, men ikke i sanntid (scenario 2):

- Folkeregisteret
 - **Endring/hendelse:** Tilbyder «legger» alle hendelser i en hendelsesliste det er en «Push» til hendelseslisten. Folkeregisteret tilgjengeliggjør hendelseslisten for konsument og hendelsene i listen har en avtalt levetid.
 - **Behov:** Konsumentene ønsker selv å styre sekvens på lesing og hvor mange hendelser konsumenten skal lese. Samme hendelser kan leses av flere systemer hos konsumentene og man kan lese hendelser så mange ganger man ønsker. Konsumentene må selv holde orden på en intern feed-peker som viser hvor langt man har lest i hendelseslisten. Det er behov for at konsument kan gjøre en «Pull» av hendelsene i hendelseslisten for å oppdatere sin tjeneste.

1.4.2 Behov for tilgangsstyring ved deling av hendelser

På samme måte som ved deling av data vil tilgang til å abonnere på gitte typer hendelser, kreve god tilgangsstyring. Aktører som kun deler data kun med de som har hjemmel til å behandle dataene, vil på samme måte også kreve at det er avtaler som ligger til grunn for å abonnere på en hendelsestype. Det er i disse tilfellene behov for at tilbyder og konsument kjenner til hverandre og inngår nødvendige avtaler. En mekanisme for hendelseshåndtering må støtte tilgangsstyring, og fellesløsningene for tillitstjenester vil spille en viktig rolle i dette.

1.5 Knytning til de overordnede arkitekturprinsippene

De overordnede nasjonale arkitekturprinsippene ligger også til grunn for en beste praksis for «Arkitektur for hendelser i felles økosystem»:

Ta utgangspunkt i brukernes behov (Prinsipp 1)

For hendelsesdrevet arkitektur ivaretas dette gjennom:

Tjenestekjeder kan utvikles organisk, og tilpasse seg brukers behov uten endring av eksisterende tjenester.

Beskrivelse:

Ved å basere tjenestene på hendelser i en aktørs livssyklus vil tjenestekjeden kunne tilpasses individuelt. En hendelse kan føre til behov for en tjeneste, tjenesten brukes og fører til endring i data. Endring i data kan føre til en ny hendelse som igjen fører til behov for en tjeneste. En tjeneste kan bruke resultatet fra en eller flere andre tjenester.

Lag digitale løsninger som støtter samhandling (Prinsipp 6)

For hendelsesdrevet arkitektur ivaretas dette gjennom:

Hendelser vil bidra til samhandlingsevne på tvers av virksomheter, sektorer, fagområder og bransjer og støtte distribuerte arkitekturer.

Beskrivelse:

Ved å basere tjenestene i en tjenestekjede på hendelser vil hver enkelt tjenestetilbyder kunne optimalisere sin tjeneste uavhengig av de andre tjenestetilbyderne. Endring i en tjeneste vil ikke påvirke de andre tjenestene i kjeden.

Standardiserte beskrivelser (Prinsipp 6, anbefaling 6.6)

For hendelsesdrevet arkitektur ivaretas dette gjennom:

Hendelser og tjenester skal beskrives på en standardisert maskinlesbar måte.

Beskrivelse:

Ved å beskrive hendelser og tjenester etter gjeldende standarder og tilgjengeliggjøre beskrivelsene i en hendelse- og tjeneste-katalog, vil andre kunne oppdage og gjenbruke tjenester i en tjenestekjede. Tjenester vil kunne tilbys basert på hendelser/resultater fra andre tjenester.

Del og gjenbruk løsninger (Prinsipp 5)

For hendelsesdrevet arkitektur ivaretas dette gjennom:

Tjenester er autonome, de kan tilbys selvstendig eller i en tjenestekjede.

Minst mulig avhengighet (Prinsipp 5, anbefaling 5.7)

For hendelsesdrevet arkitektur ivaretas dette gjennom:

Sikre at det er minst mulig avhengighet mellom tilbyder og konsument, og at tjenestene er løst koblet.

Beskrivelse:

«Konsument» og «Tilbyder» av hendelser trenger ikke vite om hverandre. Utveksling av informasjon

skjer uten direkte integrasjon. Mange konsumenter kan få tilgang til hendelser uten at tilbyder er involvert. Konsumenter kan selv kople seg opp ved å abonnere på gitte hendelsestyper (som de har tilgang til).

2 Arkitektur for hendelser i Felles Økosystem

Hendelsesdrevet arkitektur er et designparadigme der en tjeneste kjøres etter å ha mottatt en hendelse. Det gjør det mulig å utveksle informasjon mellom distribuerte løsninger på tvers av organisasjoner i tilnærmet sanntid uten komplekse og tette integrasjoner.

Hendelsesdrevet arkitektur er en kjent arkitekturstil brukt i ulike sammenhenger, og er vanlig i bl.a applikasjoner bygget med lettvekts-mikrotjenester. Denne type arkitektur er velegnet i Felles Økosystem (Økosystem for nasjonal digital samhandling og tjenesteutvikling).

«Arkitektur for hendelser i Felles Økosystem» er et grunnlag for å utveksle og agere på hendelser på tvers av virksomheter, domene og tjenester. Det er ikke en teknisk løsningsarkitektur, men vil peke på konseptet og en beste praksis for utveksling av hendelsesinformasjon slik at sammenhengende tjenester kan realiseres.

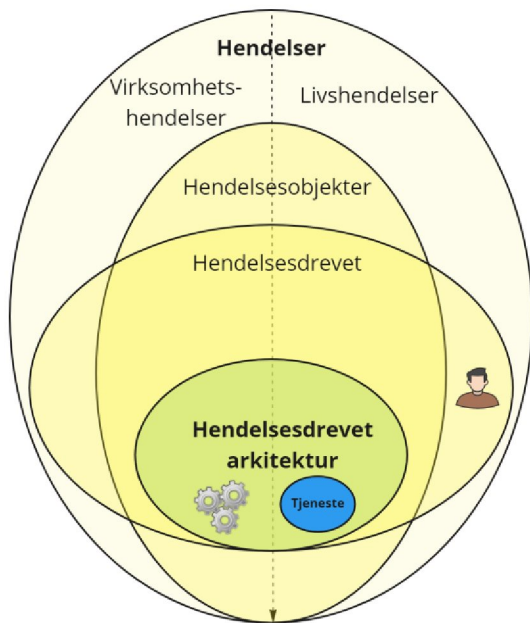
For enkelte virksomheter vil hendelsesdrevet løsningsarkitektur kunne være nyttig i eget systemlandskap, men vil måtte tilpasses virksomhetens interne prosesser og arkitektur. Vi ønsker derimot å fokusere på bruk av en slik arkitekturstil ifb. utveksling av hendelser mellom virksomheter, domene og tjenester.

2.1 «Hendelsesdrevet arkitektur» i en verden som består av hendelser

Hendelser er noe vi omgir oss med kontinuerlig. Generelt er hendelser ting som «skjer», en endring i tilstand som har oppstått. Alt som skjer i verden deltar eller bidrar således til hendelser. Mennesker og virksomheter er i sin natur hendelsesdrevet, men dette blir for generelt når vi skal lage digitale løsninger.

Digitale løsninger kan også være hendelsesdrevet, men bruker ikke hendelsesdrevet arkitektur iht dets prinsipper. En digital løsning kan tilby eller lagre et hendelsesobjekt, men blir den tilbudt i det øyeblikk hendelsen oppstod? Er det sterke bindinger mellom tilbyder og konsument?

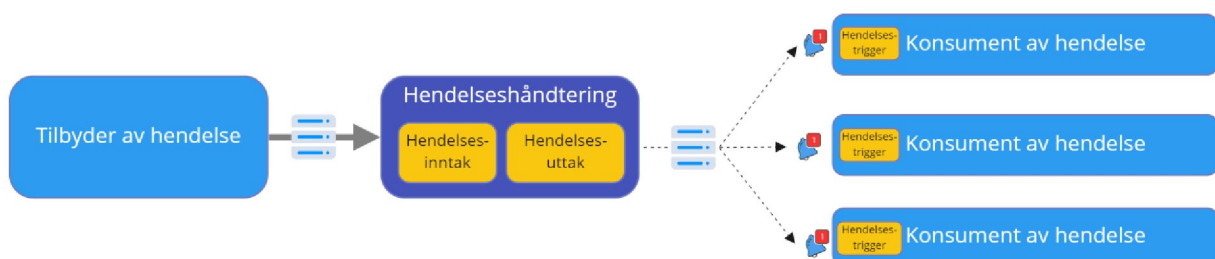
Felles prinsipper og forståelse for hendelsesdrevet arkitektur er en forutsetning for å dra nytte av denne arkitekturstilen på tvers av ulike virksomheter.



- **Hendelsesdrevet arkitektur**
 - er en arkitekturstil der *komponenter og tjenester i en teknisk løsning er hendelsesdrevet*. Tilbyder og konsument av hendelse er minimalt koplet sammen kun via hendelser, og konsument agerer på et hendelsesobjekt.
- **Hendelser**
 - Endring i en tilstand som kan utløse behov for tjenester. En hendelse er «noe som har skjedd».
- **Virksomhetshendelser**
 - Hendelser som er meningsfullt i en forretnings- eller domenekontekst.
- **Livshendelser**
 - Hendelser som er meningsfullt i en innbyggerkontekst.
- **Hendelsesobjekt**
 - Hendelse som er maskinlesbar og formidler informasjon om hvilken endring som har skjedd.
- **Hendelsesdrevet**
 - Atferden til en aktør som handler når den gjenkjenner en hendelse, fremfor å respondere på kommandoer

2.2 «Hendelsesdrevet arkitektur – logisk»

Utgangspunktet er når en hendelse oppstår i et system som er interessant for andre. «Tilbyder av hendelse» er løst koplet til «Konsument av hendelse» og de trenger ikke vite om hverandre. Det er høy grad av autonomi mellom aktørene, og konsumenter kan få tilgang til hendelser uten at tilbyder er involvert. Det kan være 1, mange eller svært mange aktører som har interesse for en hendelse.



Merk at dette er et logisk nivå. Løsningsmessig kan utveksling av hendelser gjøres på flere måter, avhengig av behov og mønstre for utveksling av hendelser.

«Tilbyder av hendelse» representerer en aktør som ønsker å publisere en hendelse på bakgrunn av endringer i en tilstand, enten dette er resultat av en tjeneste eller andre endringer i en datakilde. «Tilbyder av hendelse» publiserer eller lagrer denne endringen i en logisk arkitekturbyggekloss for «Hendelseshåndtering». En slik hendelse må tilgjengeliggjøres for en eller flere «Konsument av hendelse». «Hendelseshåndtering» innebærer ulike kapabiliteter for tilgangsstyring og tilgangskontroll utifra behov og valgt mønster for utveksling av hendelser. En «Konsument av hendelse» som skal være mest mulig løskoplet fra en «Tilbyder av hendelse» vil f.eks. ha behov for å kunne abonnere på relevante hendelser, mest mulig selvbetjent.

Hendelse

Endring i en tilstand som kan utløse behov for tjenester. En hendelse blir sendt ut i det øyeblikk det skjer en endring i tilstand hos «Tilbyder av hendelse», enten dette er endringer i data som resultat av en tjeneste eller endringer i en sammenstilt datakilde.

Hendelsesobjekt

Hendelse som er maskinlesbar og formidler informasjon om hvilken endring som har skjedd, slik at en «Konsument av hendelse» kan forstå og agere basert på informasjonen.

Tilbyder av hendelser

Når en datakilde endres, og endringen(e) er interessante for andre, kan dette tilbys som en hendelse. Hendelsen må publiseres («Push») eller lagres i «Hendelseshåndtering».

Konsument av hendelser

En tjeneste eller løsning som skal agere basert på en eller flere spesifikke hendelser som oppstår. En eller flere konsumenter handler i det øyeblikk en hendelse blir gjenkjent.

Hendelseshåndtering

Mekanisme som sikrer at hendelser er transportert og behandlet riktig mellom «Tilbyder av hendelse» og «Konsument av hendelse». «Tilbyder av hendelse» publiserer hendelsen til denne («Hendelsesinntak»), mens «Konsument av hendelse» får tilgang («Hendelsesuttak»).

Ulike behov og ulike mønstre for utveksling av hendelser krever ulike kapabiliteter i denne. Se mer om dette i kapittel om «Kap. 2.3: Hendelsesmønstre i «Arkitektur for hendelser i Felles Økosystem»

Varsel/Melding

En melding til en «Konsument av hendelse» at en spesifikk hendelse har oppstått, og inneholder et hendelsesobjekt konsumenten kan agere på.

Abonnement

De type hendelser en «Konsument» ønsker å få tilgang til. Konsumenter kan *selv* kople seg opp ved å *abonnere* på gitte hendelsestyper (som de har tilgang til).

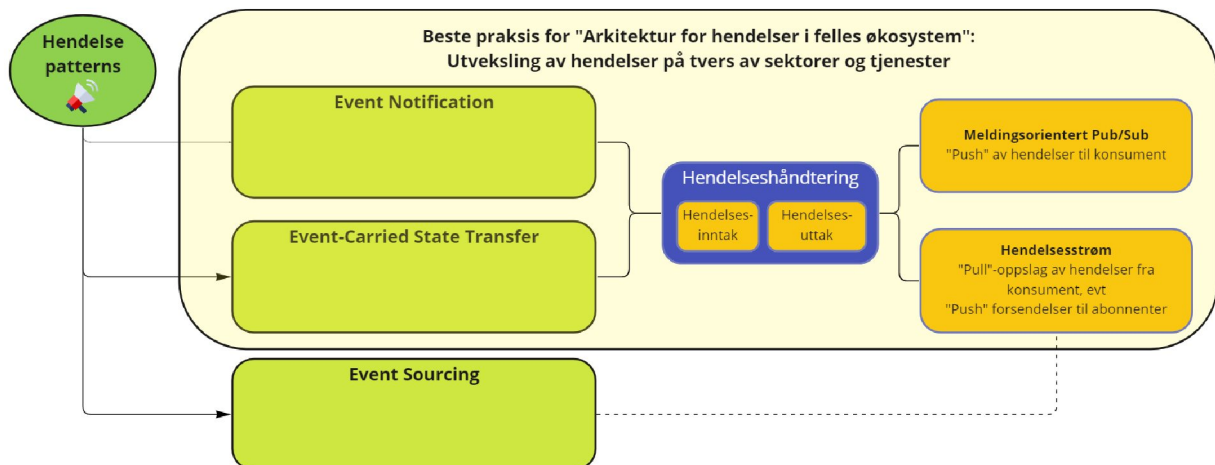
2.3 Hendelsesmønstre i «Arkitektur for hendelser i Felles Økosystem»

Det vil ikke være én måte og utveksle hendelser på i et felles økosystem. Ulike mønstre for utveksling av hendelser vil dekke ulike behov og scenario. Det vil selvsagt også være særskilte behov som ikke

nødvendigvis passer inn i et hovedmønster, som den enkelte virksomhet selv må ta stilling til. Det er allikevel 3 hovedmønster som er aktuelle å vurdere innenfor hendelsesdrevet arkitektur.

Felles for alle mønstrene er at Tilbyder av hendelse er løst koplet til selve hendeshåndteringen, og ikke trenger å kjenne til hvilke konsumenter som får tilgang til en hendelse via denne.

Martin Fowler har beskrevet disse mønstrene i en artikkel (<https://martinfowler.com/articles/201701-event-driven.html>). I tillegg vil hvert hovedmønster kunne realiseres med 2 ulike delmønstre for tilgang til hendelsene.

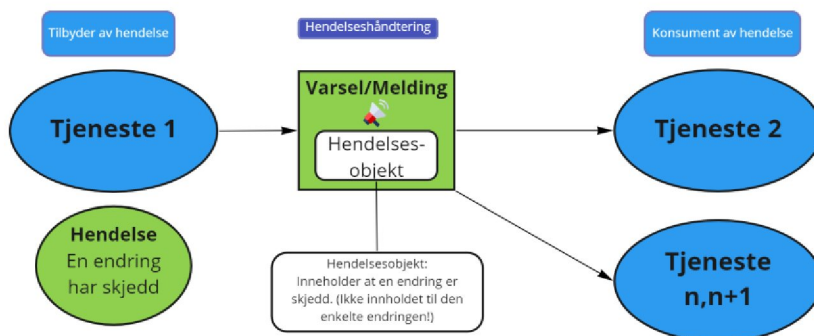


Vi har valgt å beholde de engelske navnene på hovedmønstrene i mangel av gode norske begreper.

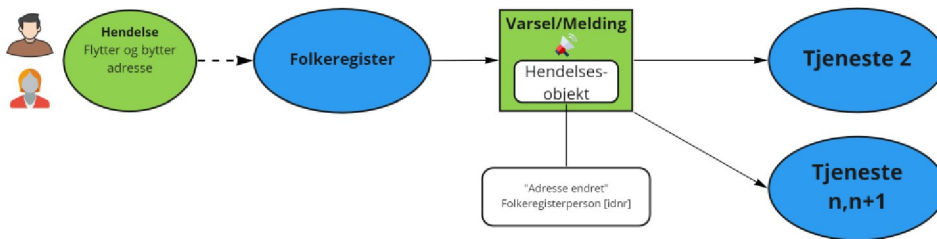
2.3.1 Hovedmønstre for utveksling av hendelser på tvers av sektorer og tjenester

2.3.1.1 Event Notification

En hendelse representerer en endring som er skjedd i data, og hendelsen publiseres som et hendelsesobjekt fra «Tilbyder av hendelse». Hendelsesobjektet inneholder kun informasjon om «hva som skjedd», ikke innholdet til den enkelte endring. Hendelsesobjektet kan f.eks. være «Adresse endret», som da vil være en hendelsestype en eller flere «Konsument av hendelse» ønsker tilgang til.



Eksempel på hendelsestype "Adresse endret" fra Folkeregister:

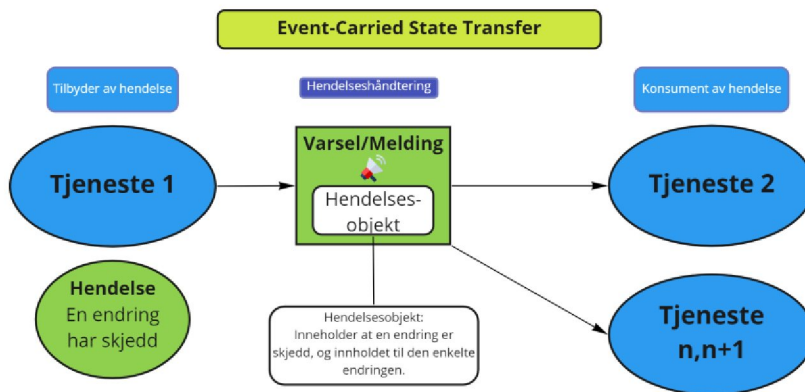


Dette er et konsistent mønster som deler minimalt med data, og som kan kombineres med oppslag i datakilde i bakkant hvis behov for mer data og ytterligere kontroll på konsumentenes tilganger.

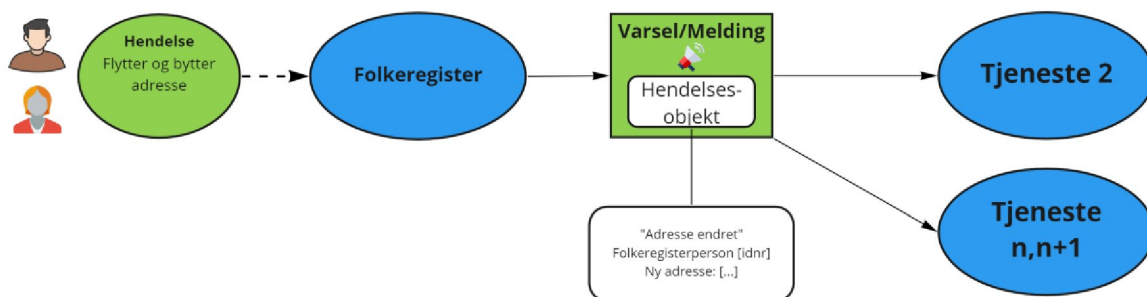
2.3.1.2 Event-Carried State Transfer

En hendelse representerer en endring som er skjedd i data, og hendelsen publiseres som et hendelsesobjekt fra «Tilbyder av hendelse». Hendelsesobjektet inneholder informasjon om «hva som skjedd», i tillegg til innholdet til den enkelte endring. Hendelsesobjektet kan f.eks. være «Adresse endret», som da vil være en hendelsestype en eller flere «Konsument av hendelse» ønsker tilgang til.

Dette mønsteret vil i mindre grad enn «Event Notification» føre til behov for oppslag i bakkant siden dataene om endringen er inkludert i hendelsesobjektet.



Eksempel på hendelsestype "Adresse endret" fra Folkeregister:



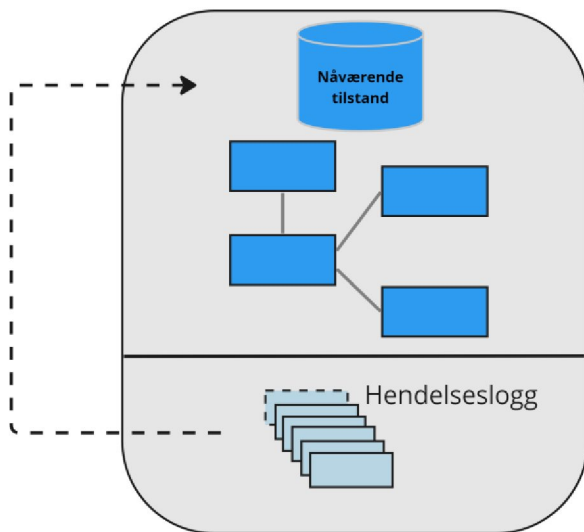
2.3.1.3 Event Sourcing

Det kan diskuteres om dette er et hovedmønster eller ikke, siden «Event sourcing» først og fremst dreier seg om at alle endringer i et system blir lagret i en egen hendelseslogg («Event store»), altså en måte å lagre data på. Det er da mulig å gjenskape et system til et bestemt tidspunkt eller tilstand basert på endringene. En tradisjonell database lagrer data i en bestemt tilstand, mens hendelsesloggen tar vare på endringene. En hendelseslogg vil være persistent til enhver tid og inneholde fakta som er såkalt «immutable», altså fakta som representerer sannhet og ikke skal kunne endres i ettertid.

Hendelseslogg er søkbar og kan inneholde øyeblikksbilder av forutbestemte tilstander. Et enkelt eksempel som belyser dette er en bankkonto, med kontotransaksjoner over tid, inkludert øyeblikksbilder pr mnd eller år.

«Event sourcing» er altså ikke direkte knyttet til «Arkitektur for hendelser i Felles økosystem» for utveksling av hendelser mellom eksterne aktører, men mer hvordan en slik hendelseslogg kan benyttes innenfor egen virksomhet.

Det er allikevel viktig å ha med denne som et hovedmønster siden virksomheter som benytter «Event store» kan tilgjengeliggjøre innhold fra denne som en «Hendelsesstrøm» for eksterne konsumenter.



2.3.1.4 Delmønstre for tilgang til hendelsene

2.3.1.4.1 Delmønster: Meldingsorientert

En hendelse blir sendt som en melding til «Konsument av hendelse», som abonnerer på aktuelle hendelsestyper. Hendelsen blir sendt som en «push» til det endepunktet en konsument har angitt. Mekanismen for hendeshåndtering må ha kapabiliteter for å sende hendelser til abonnerende konsumenter (publish-subscribe).

- Hendelse sendes til konsument i det den blir publisert av tilbyder
- Konsument av hendelse må kunne motta hendelsen i et egnet endepunkt, f.eks. som en «Webhook URL». Dette innebærer at Konsument av hendelse må implementere et endepunkt en hendelse kan sendes til.
- Hendeshåndtering må ha kapabiliteter for å sikker og garantert leveranse til riktig konsument.
 - Begrenset levetid til en hendelse siden denne i utgangspunktet fjernes når den er konsumert.
 - Må kunne håndtere situasjoner der konsument ikke er tilgjengelig, eller at feilsituasjoner oppstår
 - Abonnement på hendelser er enten selvbetjent eller kontrollert via tilgangsstyring

Mønsteret vil som nevnt innebære at hendelser blir fjernet etter en viss levetid, og nye abonnementer vil da ikke kunne få tilgang til tidligere hendelser.

2.3.1.4.2 Delmønster: Hendelsesstrøm

Vil inneholde hendelsene i en logg og støtte persistens av hendelsene som oppstår over tid. Hendeshåndtering-mekanisme trenger ikke holde på eller midlertidig lagre hendelsene siden disse vil være tilgjengelig som historikk over lengre tid. En Hendelsesstrøm har mye til felles med «Event Sourcing» men med intensjon å utveksle hendelser til eksterne aktører. En Hendelsesstrøm kan realiseres med tanke på utveksling av spesifikke hendelser som er interessante for konsumenter, og kan realiseres med enklere løsninger. F.eks. finnes det eksempler der ATOM feed er brukt til dette. (Folkeregisteret: <https://skatteetaten.github.io/folkeregisteret-api-dokumentasjon/hendelsesliste/>)

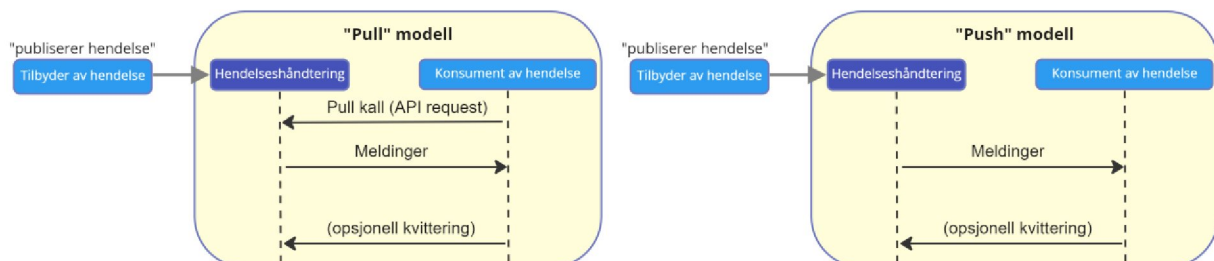
- Konsument må selv styre når en hendelse skal leses inn («pull»-modell), og implementere egen løsning for tilgang til hendelsene.
- Hendeshåndtering må ha kapabiliteter for sikker og garantert leveranse til riktig konsument.
 - Tilgang på hendelser er enten selvbetjent eller kontrollert via tilgangsstyring
- Levetiden til hendelser er lang, siden disse i utgangspunktet ikke fjernes når de er konsumert. Nye konsumenter vil kunne få tilgang til tidligere hendelser.

2.3.1.5 «Push»- eller «Pull»-modell mellom Hendeshåndtering og konsument av hendelse

I Hendelsesdrevet arkitektur er det et drivende prinsipp at man ønsker minst mulig binding mellom tilbyder og konsument av hendelser. Tilbyder av hendelse vil da publisere en hendelse («push») til en Hendeshåndterings-mekanisme med intensjon at aktuelle konsumenter vil få tilgang, uten at tilbyder må vite hvem som er konsument(er), eller hvordan og når en konsument agerer.

Dermed vil det fra konsument av hendelse sin side være ulike modeller for tilgang til hendelsen. Hvilken modell man går inn for vil bl.a. være styrt av de behov som er knyttet til tjenesten hendelsen representer, og hvilket delmønster som benyttes.

- I en «Pull»-modell må «Konsument av hendelse» selv avgjøre når en hendelse skal leses inn fra Hendeshåndtering-mekanismen. Dette kan være fra flere ganger i sekundet til f.eks. en gang pr døgn/uke osv. En slik modell vil f.eks. være egnet hvis det ikke er tidskritisk for når en hendelse blir agert på av «Konsument for hendelse». Modellen er også mest egnet i delmønster der man benytter «Hendelsesstrømmer» fremfor «Meldingsorientert».
- I en «Push»-modell vil «Konsument for hendelse» få en melding om hendelsen direkte i det øyeblikk en hendelse blir publisert til «Hendeshåndtering». Modellen er f.eks. egnet til de tilfeller det er tidskritisk at en «Konsument for hendelse» får beskjed så snart som mulig at en hendelse har inntruffet.
- I tillegg må man vurdere om det er behov for kvittering fra Konsument, altså hvilke avhengigheter man må innføre med hensyn til behov for kontroll på verdikjeden.



2.3.1.6 Virksomhetens behov vil kreve ulike realisering innenfor mønstrene

De ulike mønstrene må ses i sammenheng med varierende behov, som vil påvirke realisering i den enkelte virksomhet. «Arkitektur for hendelser i Felles Økosystem» skal først og fremst bidra til at tjenester kan utvikles, forvaltes og være sammenhengende på tvers av ulike domener, sektorer og virksomheter. Hendelsesdrevet arkitektur bidrar til løse koplinger og autonome tjenester som er en premiss for dette.

Dette er ikke en uttømmende liste, men eksempler på vurderinger som uansett må gjøres ifb. realisering av en løsningsarkitektur, men en realisering innenfor den enkelte virksomhet må ta hensyn til bl.a.:

- Sanntidsbehov og konsistens
 - Hvor tidskritisk er det at Konsument får tilgang til en hendelse i det den oppstår?
 - Det vil alltid være en viss forsinkelse fra en hendelse blir publisert til den blir lest av en eller flere Konsumenter.
- Kontroll på verdikjeden hendelsene er en del av
 - Hvilke avhengigheter er det behov for mellom aktører i verdikjeden eller tjenestekjeden?
 - Kompleksitet: Sporbarhet i verdikjeden kan bli vanskeligere i en hendelsesdrevet arkitektur, som av natur er distribuert og svært løst koplet. En aktør i en større verdikjede av autonome tjenester må vurdere behovet for evt kontroll og oversikt.
 - Evt behov for kvitteringer og statusinformasjon fra Konsument sett fra Tilbyder eller Hendelseshåndtering sin side?
- Persistens av hendelser
 - Skal Hendelser kunne benyttes senere, at den ikke forsvinner i det øyeblikk den mottas av Konsument? Konsument kan ha behov for tilgang til hendelsen i en gitt tidsperiode, f.eks. i en feilhåndteringssituasjon eller i mer forretningsmessige behov.
- Er det få, noen, mange eller svært mange aktuelle Konsumenter av en hendelse?
- Tilgangsstyring
 - Skal hendelsene kun være tilgjengelig for spesifikke Konsumenter eller ikke?
- Behov for å kombinere Hendelsesdrevet arkitektur med andre arkitekturstiler?
 - For at en Konsument skal kunne agere etter en intensjon av en hendelse kan det være behov for e-Oppslag i bakkant, spesielt hvis man bruker hovedmønster «Event notification» og det er behov for mer konsistente data.
- Kompensering
 - Er det behov for å endre på historikk eller feil i en transaksjon i en verdikjede? Vil være mer aktuell i en intern løsningsarkitektur (basert på microservices), innenfor en sektor, domene eller virksomhet, med interne bindinger. Men verdt å nevne hvis dette allikevel er aktuelt i en evt. tjenestekjede. Se: <https://microservices.io/patterns/data/saga.html>

3 Samspill i Felles Økosystem

Økosystem for nasjonal digital samhandling og tjenesteutvikling er en måte å samhandle på, og en samling verktøy og løsninger som flere kan bruke for å utvikle digitale tjenester. Dette skal bidra til at offentlige tjenester oppleves som sammenhengende for brukerne, uansett hvilken offentlig virksomhet som tilbyr dem. Hvor godt de ulike delene av økosystemet samhandler, har stor betydning for hvor gode og trygge digitale løsninger som skapes og tilbys oss som innbyggere.

Hendelser er data som må kunne utveksles mellom aktørene i Felles Økosystem, slik at vi oppnår bedre samhandling. Løse koplinger gir færre bindinger mellom aktørene, men heller ikke Hendelsesdrevet arkitektur er skånet for bindinger man må vurdere.

3.1 Teknisk Interoperabilitet

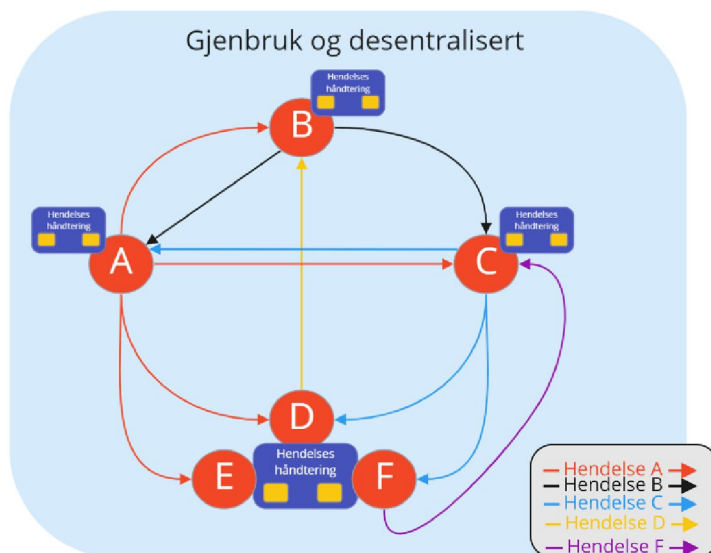
«Den tekniske evnen to eller flere løsninger eller applikasjoner har til å utveksle og bruke informasjon, evnen til å tilby og konsumere tjenester fra andre systemer, og å bruke tjenestene som utveksles slik at systemene arbeider effektivt sammen»

(<https://data.nav.no/begrep/BEGREP-540>)

Informasjon om oppståtte hendelser må kunne utveksles mellom aktørene i Felles Økosystem. Det vil være behov for mekanismer for å publisere og gi tilgang til hendelser, for både «store og små». Løsninger på tvers av virksomheter og domene må fungere sammen. I Norge har vi fellesløsninger, både nasjonale og sektorielle som kan eller skal kunne gjenbrukes.

Ulike alternative tilnærming:

1. Sentralisert arkitektur
2. Desentralisert arkitektur
3. Gjenbruk og desentralisert



Gjenbruk og desentralisert er den tilnærmingen som vil bidra til at flest mulig aktører kan bidra til at hendelser kan utveksles på tvers uavhengig av modenhet hos de ulike aktører. Her vil aktører kunne fungere i et samspill der man er enig om en felles beste-praksis. Virksomheter kan implementere egne løsninger for Hendelseshåndtering som fungerer i felles økosystem.

Et eksempel på en gjenbrukbar løsning vil være «Altinn Events». En gjenbrukbar løsning som Altinn Events er en fellesløsning som kan benyttes direkte i Altinn plattformen, eller at man gjenbruker koden og kjører den på egen plattform.

3.1.1 Utveksling av hendelser i distribuerte system

For å øke teknisk interoperabilitet for utveksling av hendelser er det behov for å se på mulige felles spesifikasjoner. Et eksempel på dette er Cloud Events-spesifikasjonen (<https://cloudevents.io/>)

Målet med CloudEvents-spesifikasjonen er å definere interoperabilitet av hendelsessystemer som lar tjenester tilby eller konsumere hendelser uavhengig av hverandre. CloudEvents-spesifikasjonen er fokusert på interoperabilitet av hendelsesformatet og hvordan det formateres, men at det kan sendes på ulike protokoller, for eksempel HTTP.

Det finnes ulike løsninger for publisering av hendelser fra ulike leverandører, og enhver virksomhet vil gjøre ulike valg for egen løsningsarkitektur utifra ulike behov, drivere og rammebetingelser. Større systemleverandører som f.eks AWS, Microsoft, Google har løsninger som publiserer hendelser i ulike formater. Dette øker behovet for en felles og normaliserende spesifikasjon ved utveksling av hendelser, slik som f.eks. CloudEvents-spesifikasjonen. Vi vil da unngå at konsumenter av hendelser må implementere særskilt logikk mot den enkelte tilbyder av hendelser.

Eksempel på et hendelsesobjekt i CloudEvents

<pre>{ "specversion": "1.0", "type": «skifteattest.utstedet», // Kan versjoneres "source": «App-application-id», "id": "9aeb0fdf-c01e-0131-0922- 9eb54906e209", "time": "2022-11-05T20:13:39.4589254Z", "dataschema": "json-schema-url", "data": { «userid»: "cust-id", "firstname": "some firstname", "lastname": "some lastname" } }</pre>	<p>Et hendelsesobjekt kan f.eks. inneholde</p> <ol style="list-style-type: none"> 1. Hendelsestype, hva slags hendelse har oppstått (type) 2. Applikasjonen som publiserte hendelsen (source) <ul style="list-style-type: none"> • Unik kombinasjon av 1 + 2 3. Globalt unik id for meldingen (id: unngår doble transaksjoner) 4. Tidspunkt for publisering (time) 5. Schema for hendelsen (En URL lenker til korresponderende JSON-schema) 6. Hendelsesdata iht til Schema (data) 7. (Evt signert hash for integritetssjekk)
--	---

3.2 Semantisk Interoperabilitet

Semantisk interoperabilitet er «evnen til å virke sammen, utveksle informasjon og til å bruke informasjonen som er utvekslet», og er viktig for deling av data, herunder hendelsesdata.

Det er under arbeid en Tjeneste- og hendelses katalog ifb arbeidet med «Felles Datakatalog» i Digdir, men denne er på et svært generisk nivå, og dekker ikke hendelser som representerer endringer i data og tilstand.

Det vil antagelig være behov for en hendelses katalog som kan beskrive hendelsestyper, og relatere disse til hendelsesdata via en informasjonsmodell. Ved å ha en katalog over hendelsestyper kan aktører finne relevante og interessante hendelser de kan agere på, og også informasjon om hvilke tjenester som resulterer i hvilke hendelser.

4 Vedlegg: Eksempel på behov og scenario

Her har vi samlet noen eksempler der hendelsesdrevet arkitektur enten allerede er benyttet eller det er behov for å benytte en slik arkitekturstil.

4.1 Dødsfall og Arv

Når noen dør utløses ulike rettigheter og plikter hos de etterlatte, blant annet:

- rett til gravferdsstønad
- videreføring av ulike pensjonsytelser
- overføring av tinglyste eiendommer
- oppgjør og avvikling av kundeforhold hos finansinstitusjoner og private tjenestetilbydere.

Dagens situasjon: Prosessene krever at en person som er gitt fullmakt av arvingene selv kontakter aktørene, fremviser en skifteattest, og fyller ut og sender inn et skjema eller på annet vis selv igangsetter en dialog med den aktuelle aktøren.

Ved hendelsesdrevet arkitektur kan disse tjenestene kunne leveres proaktivt og sammenhengende for de etterlatte:

Når domstolsadministrasjonen/Tingretten utsteder en skifteattest, kan hendelsen «Utstedt en skifteattest» og «utpekt en fullmektig» publiseres og tilgjengeliggjøres for alle bemyndige aktører. Aktørene kan dermed proaktivt agere på hendelsene og sette i gang egne prosesser. Tjenestene kan tilbys sammenhengende for de etterlatte.

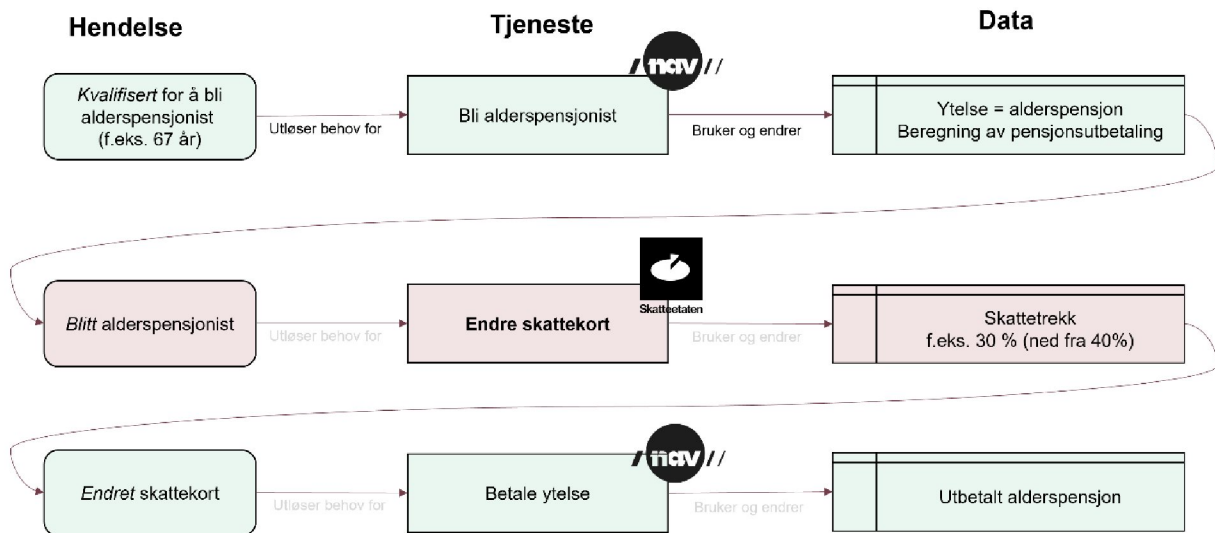
Flere hendelsestyper vil publiseres i videreutvikling av livshendelsen «Dødsfall og arv», og er grunnlag for nye, proaktive og sammenhengende tjenester. Målet er å oppnå dynamiske tjenestekjeder, også for tilstøtende livshendelser eller andre sammenhengende tjenester.

4.2 Fremtidens innkreving

Fremtidens innkreving skal blant annet samarbeide med flere andre etater om å lage helt nye måter å:

- forebygge oppbyggingen av gjeld
- kreve inn det offentliges penger
- Vi tar utgangspunkt i brukeren
- Tjenesteyting som en del av brukerens egne prosesser – brukers økosystem
- Helhetlige brukerreiser på tvers av sektorer og forvaltningsnivåer
- Hendelser utløser proaktiv dialog med brukerne
- Digital transformasjon utover organisatoriske skillelinjer
- Digitale løsninger hjelper brukerne til å gjøre rett fra start og forebygger mislighold
- Sanntidsdeling av informasjon – og også på tvers av landegrenser
- Sammenhengende tjenester på tvers av sektorer og forvaltningsnivåer ved økt deling av data

Eksempel:



Hendelsesdrevet forretningsarkitektur i Fremtidens innkreving

Er tenkt realisert gjennom forretningsreglene

- «når»-delen av forretningsregelen beskriver hendelsen
- «så»-delen formidler informasjon om hvilken tjenesteyting (handling/ behandlingsløp/aktivitet) som skal skje

4.3 Kunngjøringer fra Foretaksregisteret, Regnskapsregisteret, Partiregisteret, Konkursregisteret og Stiftelsestilsynet

Brønnøysundregistrene tilbyr abonnementstjenester for en rekke hendelser. Hendelsene er både knyttet til organisasjoner, personer og geografiske områder. Du kan blant annet abonnere på opplysning om

- endringer av navn, adresser, roller og kapital, godkjent årsregnskap, fusjon og fisjon, avvikling, konkurs, tvangsavvikling, gjeldsforhandling
- konkurs, gjeldsforhandling
- oppløsning/sletting, foretaksnavn, endringer, konkurs/tvangsavvikling, kreditorvarsel, mottatte årsregnskap, politiske partier, gjeldsforhandling, kunngjøringer fra Stiftelsestilsynet

Disse hendelsene publiseres ikke på en måte som gjør det mulig for andre tjenester å agere automatisk på dem, men her vil en hendelsesdrevet arkitektur kunne åpne for nye muligheter i fremtiden.

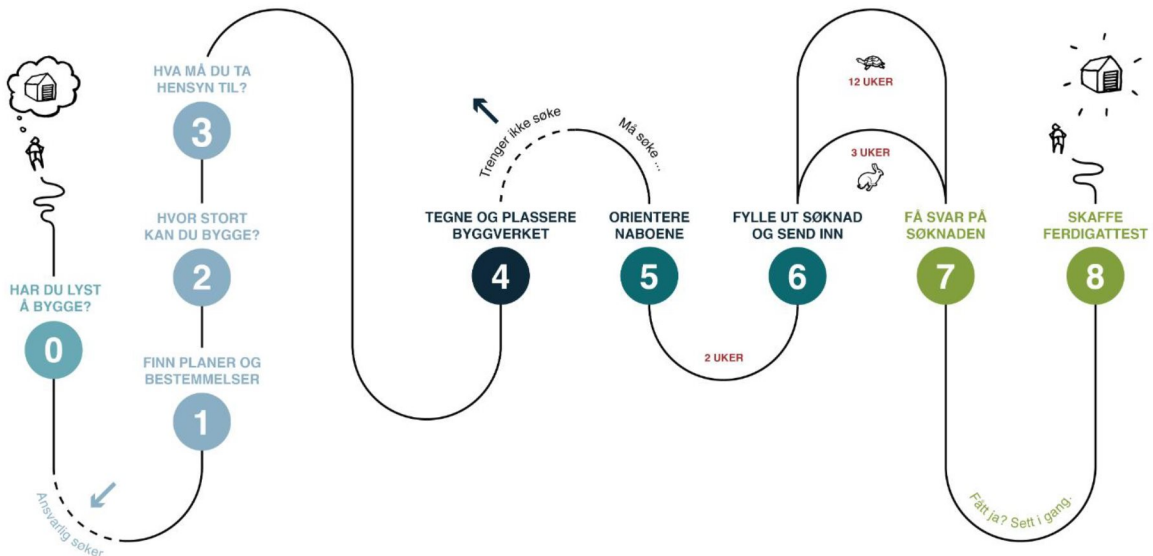
4.4 Folkeregisteret og Innbyggerkontakt

Innbyggerkontakt er tjenesten som gjør det enkelt for kommuner, fylkeskommuner og interkommunale selskaper å gi relevant informasjon til sine innbyggere når de trenger det.

Innbyggerkontakt leser hendelser fra folkeregisterets hendelsesliste og benytter kontakt og reservasjonsregisteret i forbindelse med varsling og meldinger til innbyggere. Løsningen ønsker også å benytte Matrikkelen og hendelser fra andre registre i fremtiden.

4.5 Byggesøknad

eByggesøk er en veileder som hjelper deg med både utfylling og innsending av din digitale søknad. Den guider deg igjennom en jungel av begreper og skjemaer og veileder deg helt fra du lurer på om du trenger å søke, til en ferdig innsendt byggesøknad og ferdigattest.



eByggesøk er en sammenhengendetjeneste som fremstår som orkesteret, men denne tjenesten inneholder en rekke hendelser. For eksempel «nabovarsel er sendt», «byggesøknad er sendt» og «byggesøknad er godkjent». Vi mener at denne tjenesten vil kunne benytte hendelsesdrevet arkitektur, og at det vil bidra til løsere koblinger mellom de involverte aktørene.

4.6 E-Bevis

eBevis henter informasjon fra flere datakilder om leverandører ved offentlige anskaffelser. eBevis er en orkesteret tjeneste, og sørger blant annet for at virksomhetsinformasjon fra Brønnøysundregistrene og opplysninger om restanser fra skatteetaten blir tilgjengelig i fagsystemene til hver enkelt virksomhet som bruker tjenesten. Dette fungerer fint da det er klart definerte aktører som inngår i tjenesteleveransen og alle er kjent for dirigenten (eBevis).

Løsningen baseres på at en aktør kan sende forespørslers på én eller flere datasett. Løsningen er tilrettelagt for formidling av data uten allmenn tilgang. Oppdragsgiver må ha en hjemmel til å hente informasjonen, eller innhente samtykke fra leverandøren det innhentes opplysninger om.

Aktørene (oppdragsgiverne) kan etterspørre informasjon om: arbeidsgiveravgift og mva fra skatteetaten, bemanning fra Foretaksregisteret, informasjon fra Regnskapsregisteret og Renholdsregisteret.

Vi mener denne tjenesten kan effektiviseres ved bruk av hendelser. For eksempel kan oppdragsgiver få tilgang til skatteopplysninger umiddelbart fremfor at leverandør må samtykke til oppdragsgiver om innsyn i egne skatteopplysninger. Det er behov for en «Push» av hendelsen til konsument.

4.7 Advokatregisteret og representasjon

Altinn Autorisasjon benyttes til å registrere hvem som er utnevnt til å opptre på vegne av en innbygger eller virksomhet. Et representasjonsforhold kan etableres av bruker selv eller gjennom relasjoner/representasjon i et autorativt register som vergemål- eller advokatregisteret. Tilgang til

informasjon om representasjonsforhold gjør det enklere for innbygger eller virksomhet å la seg representere ved bruk av digitale tjenester. I dag hentes og videreformidles informasjon fra advokatregisteret og vergemålsregisteret. Ved å publisere endringene, både i det representasjoner er inngått og når de av ulike grunner er opphevet, som hendelser kan den som opptrer på vegne av en innbygger eller virksomhet benytte- eller stoppe å benytte ulike tjenester i sanntid.

5 Vedlegg – Leseliste og kilder

I arbeidet har vi brukt ulike typer kilder. Bøker, artikler på Web, kurs og Webinar, blogger. Her følger en delvis oversikt over noe av bakgrunnsmateriale som er benyttet.

Bøker

- Building Microservices, 2nd Edition, Author: Sam Newman
- Event Processing: Designing IT Systems for Agile Companies, Author: K. Chandy and W. Roy Schulte
- Learning Domain-Driven Design, Author: Vlad Khononov

Blogger og artikler

- <https://martinfowler.com/articles/201701-event-driven.html>
- <https://martinfowler.com/eaaDev/EventSourcing.html>
- <https://www.confluent.io/blog/build-services-backbone-events/>
- <https://www.confluent.io/blog/journey-to-event-driven-part-1-why-event-first-thinking-changes-everything/>
- [Event-driven architecture and semantic coupling | by Prajwalan Karanjit | Towards Data Science](#)
- Cloud Events: [spec/primer.md at main · cloudevents/spec · GitHub](https://github.com/cloudevents/spec/blob/main/spec.md)

Video

- Goto 2017, Martin Fowler: <https://www.youtube.com/watch?v=STKCRSUsyPO>
- What is Event Driven Architecture, part 1-4: <https://www.youtube.com/watch?v=DQ5Cbt8DQbM>
- <https://www.youtube.com/watch?v=7Bh10yAycws>

Webinar og kurs

- Microservice fundamentals - Understanding the power and complexity of microservice architectures, Sam Newman, O'Reilly Media, Inc.
- Microservice Collaboration - Events, sagas, and schemas, Sam Newman, O'Reilly Media, Inc.
- Architecture foundations: Styles, patterns, and trade-offs - A survey of modern architecture patterns, Neal Ford, O'Reilly Media, Inc.