Bilag B: Mønstre for implementering af Overdragelse

Bilaget supplerer publikationen Vejledning i genbrug af data i selvbetjeningsløsninger med en række forskellige mønstre for overdragelse i sammenhængende selvbetjeningsforløb som indebærer, at brugeren skal anvende mere end en løsning for at klare sit ærinde med det offentlige.

Hvis et projekt, der designer eller videreudvikler et selvbetjeningsforløb, ønsker at implementere Overdragelse til et andet, efterliggende selvbetjeningsforløb, er der en række forskellige mønstre for Overdragelse, der kan være relevante.

Dette bilag gennemgår seks mønstre og diskuterer kort deres fordele og ulemper med henblik på at tjene til inspiration for projektets endelige designvalg. Læseren bør være opmærksom på, at der er tale om eksemplariske mønstre, det ikke nødvendigvis er aftalt at anvende i fællesoffentligt regi.

De seks mønstre er: 1) Stafet uden data, 2) Stafet med data, 3) Notifikation med data, 4) Notifikation uden data, 5) Kontekstlager, 6) Sammenhængende guide.

Nedenstående tabel opridser de seks mønstre. I tabellen, og i dette afsnit generelt, betegner:

SB1/SB2: Selvbetjeningsløsning 1 (foranliggende), hhv. Selvbetjeningsløsning 2 (modtager af en overdragelseskontekst)

FS1: Fagsystem 1, relateret til Selvbetjeningsløsning 1

OK: Overdragelseskontekst

Kontekstlager: Et system, der har til formål at gemme en kontekst. Er dedikeret til understøttelse af brugerrejser/overdragelse – kan evt. være en procesmotor.

Tabel 2: Oversigt og sammenligning af mulige implementeringsmønstre for Overdragelse.

Mønster Beskrivelse Brug af OK Brug af notifikation Tidsmæssigt afkoblet Robust v/ genoptagelse Mellem parter

Stafet u/ data Fuld afslutning af SB1. SB2 henter data efter once only Nej Nej Nej Nej Nej

Stafet m/ data OK sendes direkte ind i SB2 fra SB1 Ja Nej Nej Nej Nej

Notifikation m/ data Fagsystem sender notifikation med OK til bruger Ja Ja Ja Ja Ja

Notifikation u/ data Fagsystem sender notifikation uden OK til bruger Nej Ja Ja Ja Ja

Kontekstlager Ved igangsættelse af SB2 tjekkes for evt. OK gemt af SB1 Ja Nej Ja Ja Ja

Sammenhængende guide Brugercentrisk orkestrering i overordnet guide-løsning Ja Nej Nej Ja Nej

Tabel 2: Oversigt og sammenligning af mulige implementeringsmønstre for Overdragelse.

Ud over felterne ’Brug af OK’ og ’Brug af notifikation’ rummer Tabel 2 tre kolonner:

Tidsmæssigt afkoblet: Understøtter mønsteret, at brugerens gennemførsel af SB1 og opstart af SB2 kan ske i separate forløb, som fx kan betyde, at data på grund af forsinkelsen mellem de to forløb ikke længere er aktuelle?

Robust v/ genoptagelse: Er det muligt for brugeren først at påbegynde SB2 uden at afslutte, og siden hen genoptage SB2 stadig med brug af Overdragelse?

Mellem parter: En særlig form for Overdragelse går på tværs af forskellige brugere, eksempelvis hvor to forældre begge skal ind over en proces, der angår deres barn. Markeringen her angiver, om det pågældende mønster er egnet til dette.

Nedenfor gennemgås mønstrene enkeltvist.

Mønster 1: Stafet u/ data

I Figur 1 ses mønsteret for Stafet uden data. ”Stafet” dækker over, at ”stafetten gives videre” mellem de to selvbetjeningsforløb. De indgående trin er:

SB1 afgiver data til fagsystemet, som gemmer alle data som en del af Kernen. Data valideres i forbindelse med, at de gemmes, og gøres umiddelbart tilgængelige for SB2 og andre anvendere.

SB1 sender brugeren videre til SB2 – direkte i browseren, evt. via link, efter Afrundingen

SB2 henter de relevante data fra de underliggende datasæt

Figur 15.jpg

Figur 15: Stafet u/ data.

Der er i dette mønster ingen overdragelseskontekst i spil. På den vis kan man sige, at dette mønster er et grundmønster, der giver mulighed for, at SB2 kan genbruge data, men hvor data som en del af SB1 er blevet lagt ned de rigtige steder og dermed kan hentes via ’almindelig’ once-only.

Fordelene i dette mønster er, at det er simpelt, og at der ikke er uverificerede data (i form af overdragelseskonteksten) i spil. Processen er derudover godt afkoblet – brugeren kan vælge at gå videre med forløbet i SB2, og det er samtidig muligt at genoptage SB2 senere.

Aftalemæssigt er det her vigtigt at partnerne er enige om, hvor data lagres og kan hentes til genbrug.

Mønster 2: Stafet m/ data

Dette mønster rummer blot et enkelt trin:

SB1 sender ”stafetten” videre til SB2, i form af overdragelseskonteksten, der rummer data, der kan benyttes i SB2. Brugeren fortsætter umiddelbart den overordnede proces i SB2 i samme browser/arbejdsgang.

Figur 16.jpg

Figur 16: Stafet m/ data.

Teknisk kan dette mønster implementeres med en beriget URL, hvor datafelter fra overdragelseskonteksten angives som parametre. Alternativt kan cookies i browseren bære overdragelseskontekst videre. Sikkerhedsmæssigt er dette dog ikke en optimal løsning og bør ikke benyttes ved følsomme data.

En letvægtsløsning, der kan overføre procesdata i form af, hvilken selvbetjeningsløsning brugeren kommer fra, kan være at benytte et særskilt end-point på SB2-løsningen. Konceptuelt skal SB1 sende brugeren videre til et end-point af formen https://www.sb2-tld.dk/sb1/, hvilket fortæller SB2, at brugeren kommer fra SB1. Dette kræver minimal indsats hos SB2 og er også nemt implementeret hos SB1, men kræver naturligvis, at SB1 og SB2 laver en aftale om integrationen, da SB2 nu skal vedligeholde et separat end-point kun af hensyn til SB1.

Fordele ved dette mønster er, at det er simpelt og ikke involverer systemer ud over SB1 og SB2. Ulemper ved mønsteret er, at det ikke er robust. Lukker brugeren browseren ned efter at være sendt videre til SB2, og genoptager brugeren senere SB2-forløbet på ny, er overdragelseskonteksten tabt. SB2 kan dermed ikke længere personalisere løsningen/forudfylde data baseret på overdragelseskonteksten. Brugeren skal selv finde tilbage til SB2 og vil i øvrigt få en anden oplevelse, hvis SB2 startes forfra uden overdragelseskontekst.

Mønster 3: Notifikation m/ data

I Figur 3 ses et mønster, hvor der introduceres en asynkron notifikation. De indgående trin er:

SB1 sender i Kernen sine data inklusive resultat/valg fra selvbetjeningsforløbet til fagsystemet FS1. Data indeholder den overdragelseskontekst, der er relevant for SB2.

FS1 kan herefter foretage yderligere processering af data, fx en manuel godkendelse.

Når data er færdigprocesseret, sender FS1 en notifikation til brugeren, der indeholder overdragelseskonteksten. Notifikationen kan fx tage form af Digital Post[47], en e-mail eller en SMS.

Brugeren agerer på notifikationen og tilgår SB2, fx gennem et beriget link i form af en URL med overdragelseskontekstens data angivet som parametre.

Figur 17.jpg

Figur 17: Notifikation med data

Fordelen her er bl.a., at anvendelse af en notifikation gør mønsteret velegnet til at understøtte forløb, hvor en forsinkelse i den tværgående proces ikke kan undgås – med andre ord, hvor det ikke er muligt for brugeren at gennemføre hele processen i én arbejdsgang. Notifikationen gør det nemt at genoptage processen og lade den fortsætte i SB2. Ydermere er genoptagelsen robust – en bruger kan genoptage SB2 flere gange via notifikationen, hver gang med et ensartet forløb og ud fra den samme overdragelseskontekst.

Ulemper ved mønsteret er bl.a., at den tidslige afkobling indebærer en risiko for, at data i overdragelseskonteksten mister deres aktualitet. Der skal i SB2 dermed indføres regler for, hvor længe og under hvilke forudsætninger, man ønsker at stole på data fra overdragelseskonteksten. Ligeledes er mønsteret ikke velegnet, hvis der indgår følsomme data i overdragelseskonteksten.

Mønster 4: Notifikation u/ data

I lighed med forrige mønster indgår der her en notifikation, men i denne variant bærer notifikationen ingen data/overdragelseskontekst. De indgående trin er:

SB1 sender i Kernen sine data inklusive resultat/valg fra selvbetjeningsforløbet til fagsystemet FS1.

FS1 kan herefter foretage yderligere processering af data, fx en manuel godkendelse. Når data er færdigprocesseret, gemmes alt data i de relevante datasæt (fx registre), hvorfra de med det samme gøres tilgængelige for genbrug.

FS1 sender en ikke-databærende notifikation til brugeren

Brugeren benytter link i notifikationen til at tilgå SB2

SB2 henter de relevante data, som FS1 har gemt, via once only fra de bagvedliggende datasæt

Figur 18.jpg

Figur 18: Notifikation uden data

Fordelene i dette mønster er mange. Det understøtter afkobling i tid mellem selvbetjeningsforløb, der indgår i den overordnede, tværgående proces. Notifikationen gør, at SB2 kan genstartes af brugeren. Og det, at SB2 (i lighed med mønster 1) benytter sig af de relevante datasæt direkte og dermed undgår kompleksiteten i at håndtere de uvaliderede og potentielt ikke-aktuelle data i en overdragelseskontekst, mindsker kompleksiteten og mulige fejlkilder.

Ulempen ved mønsteret er, at det ikke understøtter en fuld afvikling af den tværgående proces i én arbejdsgang. Brugeren vil skulle vente på notifikationen, der ankommer ad en separat kanal, før end han/hun meningsfyldt kan tilgå SB2. Tilgår brugeren alligevel SB2, er der en risiko for, at SB2 arbejder ud fra ikke-opdaterede data i de bagvedliggende datasæt, hvis data fx afventer manuel godkendelse i FS1.

Mønster 5: Kontekstlager

En mulig løsning for at facilitere Overdragelse er at introducere en ny arkitekturkomponent – et Kontekstlager – der har til opgave at holde overdragelseskontekster mellem systemer.

Kontekstlageret kan implementeres i sammenhæng med enten SB1, med SB2 eller som en selvstændig, central komponent, der for den sags skyld kan betjene flere selvbetjeningsløsninger end blot SB1 og SB2.

Trinene i dette mønster er:

SB1 gemmer overdragelseskontekst i Kontekstlager

Bruger igangsætter efterfølgende SB2. Dette kan enten ske i samme arbejdsgang for brugeren, eller tidsligt afkoblet

SB2 forespørger (forud for sin Forberedelsesdel) på Kontekstlageret, om der findes en overdragelseskontekst fra SB1, der hører til den aktuelle bruger. Hvis ja, hentes denne og anvendes i SB2-forløbet.

Figur 19.jpg

Figur 19: Kontekstlager.

En fordel ved dette mønster er, at det potentielt kan understøtte Overdragelse fra flere forskellige selvbetjeningsløsninger. Dette kan være relevant, hvis SB2 opfylder et specifikt men bredt anvendelige forretningsformål, der indgår som en brik i mere end én tværgående proces. Ulempen ved dette er dog, at det vil bidrage til at øge kompleksiteten i SB2s håndtering af overdragelseskonteksterne, hvor der nu ikke blot skal kvalitetstjekkes på én kontekst, men også implementeres regler for, hvilke foranliggende selvbetjeningsløsninger, der har forrang; i hvilket omfang, flere overdragelseskontekster kan kombineres, m.m.

En anden ulempe er, at den nye komponent i sig selv bidrager til at øge kompleksiteten i systemlandskabet. Det skal derudover aftales mellem SB1 og SB2, hvordan overdragelseskontekster håndteres, herunder hvor længe Kontekstlageret skal opbevare dem, inden de slettes. I forhold til datasikkerhed er der også udfordringer i at lagre data i overdragelseskontekster et nyt sted. GDPR taler fx om, at borgere kan have ret til at få slettet data om dem selv. Hvis personhenførbare data indgår i overdragelseskonteksten og dermed i Kontekstlageret, skal der defineres regler om, hvilket system og hvilken myndighed, der er ansvarlig for at gennemføre en sådan sletning i Kontekstlageret.

Mønster 6: Sammenhængende guide[48]

Dette mønster dækker over det, man kunne kalde en top down-anvendelse af Overdragelse. Her sigter man mod at implementere en overliggende guide-løsning til at understøtte en brugers tværgående proces/brugerrejse. Som et led i denne guide kan funktionalitet i eksisterende selvbetjeningsløsninger genbruges. Dette gør koordineringen i Overdragelsen langt mere styret, da den så at sige sker oppefra og ned, frem for horisontalt mellem selvbetjeningsløsninger, som tilfældet har været i de forudgående fem mønstre.

De indgående trin i dette mønster, som vist i Figur 6, er:

Brugeren anvender en brugercentrisk orkestreringsløsning til at understøtte en tværgående proces. Undervejs i dette orkestreringsforløb (der i sig selv er en selvbetjeningsløsning og indeholder en selvbetjeningskonteskt) sendes brugeren ind SB1, der er en eksisterende selvbetjeningsløsning. I denne sammenhæng dannes den relevante overdragelseskontekst OK1.

Den overordnede guide-løsning har, hvor det er muligt og fordelagtigt, mulighed for at trække direkte på forretningsservices udstillet af de relevante, bagvedliggende fagsystemer. Dette kan være hensigtsmæssigt for at skabe det mest brugervenlige forløb, hvor brugeren bliver i den samme løsning (guide-løsningens brugerflade) og ikke skifter ind i en separat selvbetjeningsløsning, der kan have sit eget look and feel.

Figur 20.jpg

Figur 20: Sammenhængende guide.

Fordelen ved dette mønster er først og fremmest, at det introducerer en separat komponent (guide-løsningen), der har til formål at skabe en god, sammenhængende oplevelse for brugeren på tværs af eksisterende løsninger. Guiden har også mulighed for at sikre, at de overdragelseskontekster, der indgår i forløbet, er aktuelle.

Ulempen er, at mønsteret kræver introduktion af endnu en komponent i form af guide-løsningen, der både skal kunne agere selvbetjeningsløsning samt virke som procesafviklings- og orkestreringsplatform.

En mulig udvidelse af dette mønster er at anvende RPA/robotics-teknologi til automatisk at til at gennemføre fx SB1 i skitsen i Figur 6. Hvis dette gøres serverside, får man mulighed for at pakke SB1-løsningen fuldstændigt ind og effektivt set udstille den som en service, som orkestreringsløsningen kan kalde i forløbet. Det kræver, at guide-løsningen først indsamler alle data fra brugeren, som brugeren ellers ville skulle indtaste i SB1 for at kunne gennemføre SB1-selvbetjeningsforløbet. En ulempe er derudover, at man får skabt en tæt binding til den specifikke version af SB1.

En variant af dette mønster er, at der ikke er tale om en sammenhængende guide med klart defineret start- og sluttilstand, men i stedet tale om en platform/overbliksløsning for løsning af gentagne opgaver, som fx virk.dk er for visse medarbejdere. De underliggende selvbetjeningsforløb (SB1, SB2 m.fl.) vil da kunne notificere (overdrage til) den overordnede platform via fx et Publish/Subscribe-mønster. Ligeledes vil platformen kunne fange forretningshændelser fra fagsystemer m.v. og give brugeren mulighed for at fortsætte eller igangsætte nye selvbetjeningsforløb.