DD130 - SA-Systemer - Tværgående

Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse	Status
1.0	24-11-2020	Mads Rangholm	Dokument oprettet	Planlagt
1.0	14-01-2021	Mads Rangholm	Dokument færdiggjort og internt godkendt.	Færdigt
1.0	15-02-2022	Zaki Hussain	Mindre styling rettelser.	Reviewet

Indholdsfortegnelse

- Introduktion
- Tekniske designs
  - Udgående data-opdateringer
  - Indkommende data-opdateringer
    - Request-coordination
    - EndPoint/Service opdeling

Introduktion

Dette dokument beskriver de tværgående tekniske designs som er anvendt i udstillingen af SA-services. Der beskrives hvordan valideringer udføres og hvordan data opdateres på et detaljeret niveau, som vil give en teknisk resurse et godt udgangspunkt til at vedligeholde integrationerne.

Tekniske designs

Dette afsnit beskriver de forskellige tekniske tilgange til SA-services. Der er en stærk opdeling mellem services, hvor SA-systemerne henholdsvis pusher (indkommende) eller modtager (udgående) data.

Udgående data-opdateringer

Dette sker gennem SyncHentopsamledeData, samt operationer som tillader at hente enkelte entiteter baseret på deres id ("GET-operationer").

Indkommende data-opdateringer

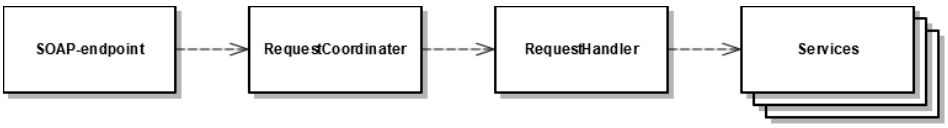
Denne sektion beskriver tekniske designs for de services, som modificerer vores data via operationer.

Der er anvendt to forskellige tilgange på tværs af alle indkommende SA-services, fordelt som vist nedenfor:

Service	Design-mønster
SyncHold	Endpoint/Service opdeling
SyncLokationer	Request-coordination
SyncSkoledagskalendere	Request-coordination
SyncTilmeldinger	Request-coordination
SyncTilstededage	Request-coordination

Request-coordination

Dette implementeringsmønster vedrør data, som er meget uafhængigt at anden data, hvor valideringer i høj grad vedrør den samme type data og ikke mere komplekse data-sammenhæng. Mønsteret tager udgangspunkt i SOAP-requestet, hvor der indhentes data via services, fra voksenuddannelses datagrundlag og validering og behandling foregår i en proces.



Logisk enhed	Beskrivelse
SOAP-endpoint	Selve modtagelsen og afsendelsen af SOAP-strukturer og repræsentering af SOAP-operationer. Mapper request-strukturen til en intern struktur, internt kaldt "validationModel" hvis klasser er postfixet med "VM". Denne model overdrages til request-coordinator.
RequestCoordinator	Koordinerer behandlingen af et request. Sørger for at modtagen data bliver behandlet i den rigtige rækkefølge og at der rapporteres korrekt på evt. opståede valideringsfejl.
RequestHandler	Håndterer behandlingen af en handling vedr. en enkelt entitet, f.eks. "Update" af en tilmelding. Der indhentes data fra voksenuddannelse's data-grundlag og denne data, kombineret med data fra request-entiteten. Der udvælges relevante valideringer, som køres på data og både evt. fejlrapportering og opdatering af data-grundlag, via service-klasser, håndteres i dette flow.
Services	Benyttes både til at indsamle data som indgår i et samlet valideringsforløb for en eller flere entiteter og til at opdatere data-grundlaget såfremt alle valideringer går godt.

EndPoint/Service opdeling

Dette implementeringsmønster vedrør opdelt ansvar ml. varetagelsen af integrationen og varetagelsen af applikationens data-grundlag og er særligt velegnet hvor der er mange data-afhængigheder, som f.eks. SyncHold. Dette sker i forskellige applikationslag som refereres til i dette dokument til som Endpoint / Integrationslag og Service-lag.

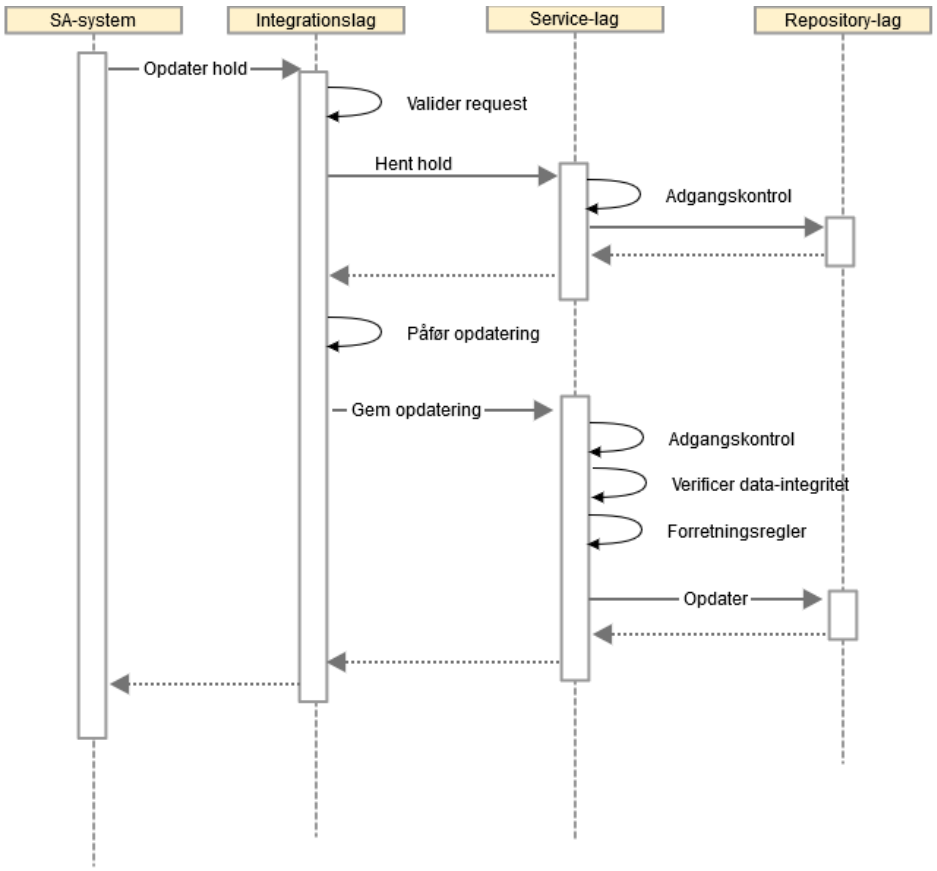
Kort Opsummering:

Applikationslag	Ansvarsområde

Integration / Endpoint	• Modtagne requests er gyldige
	  ◦ Skema (sker automatisk)
	  ◦ Ugyldige udfyldelser af skema ift. forretningsanvendelse
	  ▪ Aftalte konventioner ift. udfyldelse af data-struktur
	  ▪ Uventede koder (f.eks. en ukendt status-kode)
	• Afsendte requests er gyldige
	  ◦ Skema (sker automatisk)
	  ◦ Korrekt angivelse af overordnet fejlkode, samt fejlkoder på entitetsniveau

Service	• Adgangskontrol
	  ◦ Data kan ikke tilgås eller ændres hvis der ikke er et tilhørsforhold til identiteten af anvenderen
	• Forretningregler
	  ◦ Forretningskrav sikres opretholdt før hver ændring i data.
	  ▪ F.eks. må et hold ikke ændrer periode, så en eksisterende tilstededag falder udenfor perioden.
	• Data-integritet
	  ◦ Sammenhæng i data-grundlaget skal opretholdes. Referencer til anden data bekræftes og ved sletning verificeres det at der ikke er anden data, som refererer til det slettede forud for sletningen.
	• Applkationsspecifik fejlrapportering
	  ◦ Servicen er ikke målrettet integrationen, men systemet, og rapporterer derfor fejl gennem applikationsspecifikke exceptions, som endpointet efterfølgende kan oversætte til fejlkoder aftalt i integrationens snitflade

Eksempel på opdatering af hold:


SA-system
Integrationslag
Service-lag
Repository-lag
Opdater hold
Valider request
Hent hold
Adgangskontrol
Påfør opdatering
Gem opdatering
Adgangskontrol
Verificer data-integritet
Forretningsregler
Opdater