

# DD130 - HentUdbud

## Versionshistorik

Version	Dato	Ansvarlig	Beskrivelse	Status
0.1	?	?	Flere informationer er blevet opbevaret her	Kladde
0.2	10-12-2020	<a href="#">Mads Rangholm</a>	Tilføjet beskrivelse af batchjobbet, som fylder initial data i cachen	Kladde
0.3	15-12-2020	<a href="#">Christopher Victor Holst Nielsen</a>	Opdater hele dokumentationen i forhold til integrationer og implementation	Kladde
0.4	18-12-2020	<a href="#">Christopher Victor Holst Nielsen</a>	Opdater hele dokumentationen i forhold til integrationer og implementation	Kladde
0.5	29-12-2020	<a href="#">Christopher Victor Holst Nielsen</a>	Mindre tekstrettelser	Kladde
0.6	08-01-2021	<a href="#">Dennis Bøgelund Olesen</a>	Opdatering af dokumentation for beregning af pladser, mindre tekst rettelse og oprydning	Klar til review
1.0	18-01-2021	<a href="#">Mads Rangholm</a>	Internt godkendt	NC godkendt

## Relateret dokumentation

[D0180 - HentUdbud](#)

## Indholdsfortegnelse

- [Introduktion](#)
- [Overblik](#)
- [Tilføjelse af modul](#)
- [HentUdbud](#)
  - [Datamodel](#)
  - [Definition af webservicen](#)
  - [Håndtering af ændringer til hold](#)
  - [Dataindsamling af holdet/holdene](#)
    - [Udvid Kursuskatalog services](#)
    - [Integration af Kursuskatalog services](#)
    - [Integration af tilmelding service](#)
    - [Beregning af Holdpladsliste](#)
  - [Tilføjelse af HentUdbud SOAP service](#)
    - [Request](#)
    - [Response](#)
- [HentOptagedePladser](#)
  - [Datamodel](#)
  - [Definition af webservicen](#)
  - [Håndtering af ændringer til tilmeldinger](#)
  - [Dataindsamling for holdet/holdene](#)
  - [Beregning af FagpladsListe](#)
  - [Tilføjelse af HentOptagedePladser SOAP service](#)
    - [Request](#)
    - [Response](#)
- [Initial cache-opbygning](#)

## Introduktion

Dette dokument har til formål at beskrive det detaljerede design af servicen HentUdbud. Denne service benyttes af, bl.a. skoler, til at hente udbudte hold.

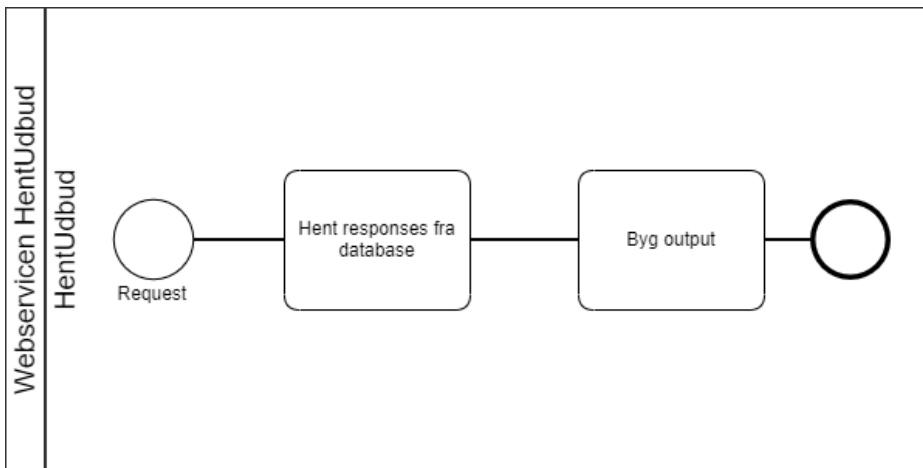
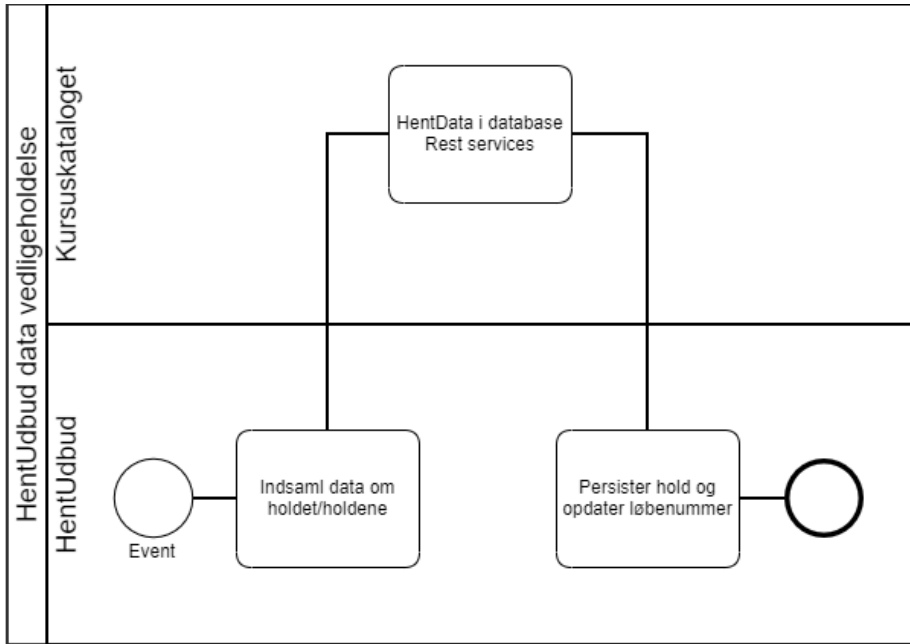
Dette gøres gennem to webservices, som begge udbydes af STIL: HentUdbud og HentOptagedePladser.

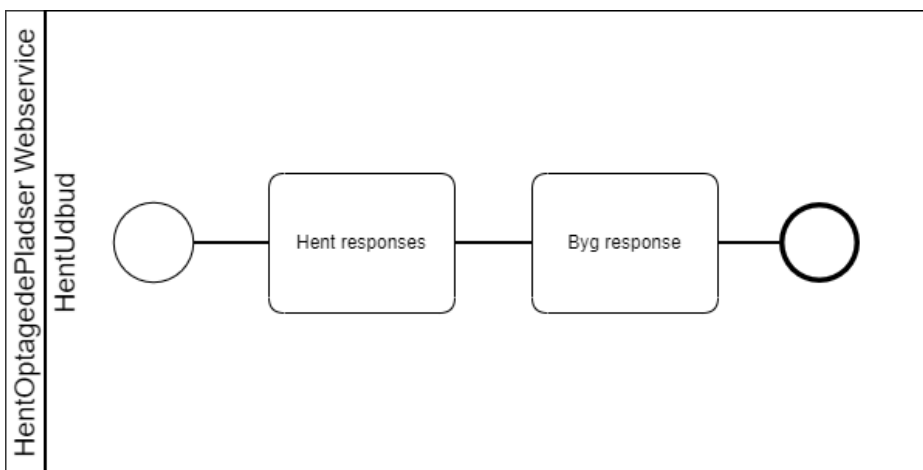
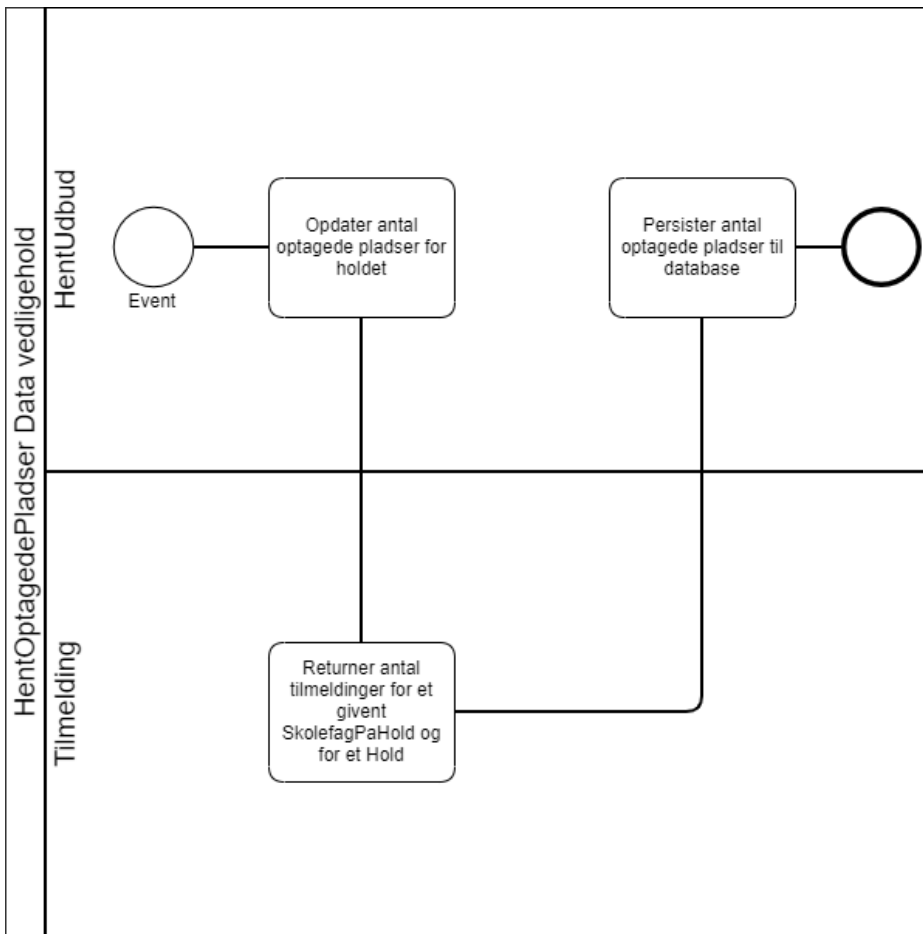
## Overblik

HentUdbud udstiller en SOAP service til at lave opslag på ændrede hold gennem en liste af DS-numre og en valgfri angivelse af om opslaget skal filtrere på indgange nyere end et specificeret nummer. Dette gøres gennem en optimeret kø med hold-data, som opdateres ved events relevant for hold fra Kursuskatalog.

HentOptagedePladser udstiller en SOAP service til opslag af optagede pladser for hold samt skolefag på hold gennem en liste af AktiGuid. HentOptagedePladser vedligeholder en tabel af antal optagede pladser for hold som opdateres ved relevante events vedrørende tilmeldinger.

Diagrammer af disse to webservices ses nedenfor:





## Tilføjelse af modul

HentUdbud og HentOptagedePladser er afkoblet fra tilmelding ved at give det sit eget modul, *hentudbud*, som har følgende submoduler:

- hentudbud (hentudbud-parent)
  - hentudbud-batch
  - hentudbud-common
  - hentudbud-database
  - hentudbud-integration
  - hentudbud-model
  - hentudbud-service
  - hentudbud-webservice
  - hentudbud-webservice-interface

## HentUdbud

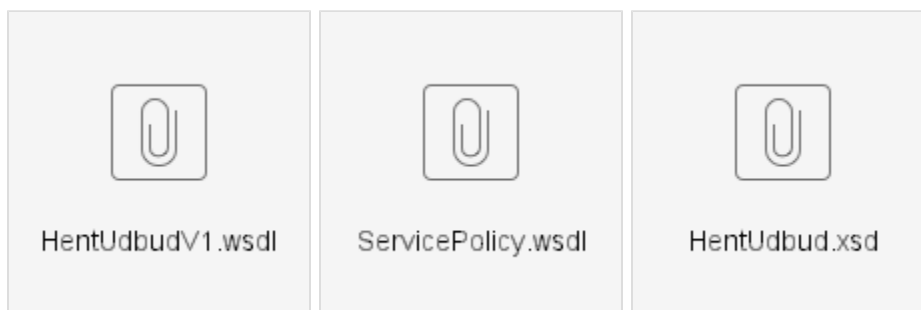
Denne sektion beskriver det detaljeret design for HentUdbud webservicen.

### Datamodel

Se [D0120 - Fysisk Datamodel - HentUdbud](#)

### Definition af webservicen

HentUdbud webservicen defineres ud fra to WSDL-filer, hvor de nødvendige endpoints og objekter til at bygge en output struktur specificeres en yderligere XSD-fil. Disse filer bliver tilføjet til submodulelet *hentudbud-webservice-interface* for at kunne bruge de genererede filer i alle submoduleler. Disse WSDL-filer samt tilhørende XSD-fil ses nedenfor:



### Håndtering af ændringer til hold

Som beskrevet ovenfor skal HentUdbud vedligeholde en kø af ændringer til hold. Denne vedligeholdelse består i, at opdatere elementer i tabellen for HentUdbud ved relevante events vedrørende hold fra Kursuskatalog.

Her udvides Kursuskataloget til at håndtere opdatering af følgende objekter:

- amu\_hold
- amu\_institution
- amu\_lokation
- amu\_uddannelse
- amu\_hold\_plads
- amu\_tillaegs\_pris
- amu\_institutionfag\_paa\_hold
- amu\_institutionfag
- amu\_uvm\_fag
- amu\_fag\_periode
- amu\_fag\_plads

Hertil tilføjes en observer, *HentUdbudDataObserver*, som håndterer disse events. For hver event vurderes der om det er relevant for HentUdbud i forhold til de felter der sættes og sendes tilbage fra webservicen (se [D0180 - HentUdbud](#)).

For hver opdatering på ovenstående objekter skal den modtage eventet, hente data omkring holdet samt ovenstående objekter og gemme denne data som tekst i kolonnen *data* i den nyoprettede database tabel.

Der indsættes et element ved at skelne mellem oprettelse af *amu\_hold* samt opdatering og sletning af selve *amu\_hold* eller relevante objekter. Ved oprettelse af *amu\_hold* indsættes et nyt element i "hentudbud\_queue", hvor kolonnen *handling* sættes som OPRET. Ved opdatering og sletning af *amu\_hold* samt opdatering af relevant objekter fjernes det tilsvarende element i tabellen og der tilføjes et nyt med opdateret løbenummer. Ved opdateringer sættes kolonnen *handling* til OPDATER og ved sletning sættes den til SLET.

## Dataindsamling af holdet/holdene

For hvert event skal det være muligt at finde alle hold der er påvirket. Her oprettes en service, der kan hente hold fra ovenstående. *HentUdbudDataService* henter hold påvirket af det specifikke event, baseret på en liste af *AktiGuid*.

## Udvid Kursuskatalog services

For at hente hold påvirket af event'et opdateres hold servicen *AmuHoldService* for Kursuskatalog REST service til at medtage mere data om hold, uddannelse samt medtage Holdpladser og Tillægspriser.

*InstitutionFagPaaHold* REST service udvides også med at hente data om *UvmFag*, *GodkendtSkole*, *GennemforendeSkole*, *Uddannelse*, *Udlisteret* til og *FagPerioder*

## Integration af Kursuskatalog services

Hentning af hold og tilhørende information gøres via *AmuHoldService* og *TilmeldingRestService*. *AmuHoldService* bruges til at hente hold med data vedrørende selve Kursuskataloget og *TilmeldingRestService* bruges til at hente hold med data vedrørende tilmeldinger. Disse services integreres i *HentUdbud* i *hentudbud-integration* submodulet.

Integrationen udføres gennem generering af nødvendige klasser og objekter ved at bygge klientet med Swagger, specificeret i pom.xml filen for *hentudbud-integration*.

Derudover tilføjes en yderligere service til at interagere med integrationen og kalde REST services, henholdsvis *KursuskatalogService*.

## Integration af tilmelding service

For at kunne bruge REST service for tilmelding som er beskyttet med at Basic Authentication lag skal integrationen sende authentication med i hvert kald til servicen i *TilmeldingService*. Dette gøres først og fremmest ved manuelt at oprette en ny bruger til system rollen ('role\_system') med et SQL script:

### Oprettelse af HentUdbud bruger

```
INSERT INTO veu_meta_info.users(username, password, salt, iteration_count) VALUES ('hentudbud',
'<password>', '<salt>', 10);
INSERT INTO veu_meta_info.user_roles(user_id, role_id)
VALUES (currval(pg_get_serial_sequence('veu_meta_info.users', 'id')),
currval(pg_get_serial_sequence('veu_meta_info.roles', 'id')));
```

Derefter tilføjes brugernavn og password til vores keystore vault, således det er beskyttet og kan bruges på tværs af kodebasen.

Ved integration af services tilføjes der en *ApiProducer* i integrations submodulet, som sætter og definerer klienten til både tilmelding og Kursuskatalog REST service. Når tilmelding REST servicen defineres defineres, sættes der authentication på *ApiClient* klassen:

### Definition af API klienter

```
@PostConstruct
public void init() {
    tilmeldingApiClient = new dk.stil.hentudbud.integration.tilmelding.ApiClient();
    tilmeldingApiClient.setUsername(hentUdbudSystemUsername);
    tilmeldingApiClient.setPassword(hentUdbudSystemPassword);
    tilmeldingApiClient.setBasePath(tilmeldingBaseUrl);

    kursuskatalogApiClient = new dk.stil.hentudbud.integration.kursuskatalog.ApiClient();
    kursuskatalogApiClient.setBasePath(kursuskatalogBaseUrl);
}
```

Derudover tilføjes configuration properties for at bruge brugernavnet og password på tværs af modulet. Dette gøres ved at tilføje en ny klasse, *ConfigurationEnum*, til submodulelet *hentudbud-common*, og definere en variabel med typen enum. Her tilføjes der yderligere .properties filer til alle miljøer i en config mappe, hvori værdierne til de forskellige properties sættes.

## Beregning af Holdpladsliste

Der udregnes optagede pladser for både *HoldPladsListe* og *SkolefagPaHoldList*.

For at udregne optagede pladser på *HoldPladsListe* bruges *TilmeldingSkoledagskalendereWebServiceDto* og *TilmeldingMedFagTilmeldingerResponse*.

For at finde datoerne der skal oprettes skal følgende gøre sig gældende for datoerne:

- Datoen er indeholdt i en skoledag på skoledagskalenderen for holdet.
- Datoen er i perioden for holdet.

Antallet af optagende pladser beregnes som antallet af tilmeldinger på holdet som overlapper datoerne.

Der kan altså godt være færre optagede pladser nogle dage end andre.

Dette er muligt ved åbent værksted, sene tilmeldinger og dispensationer.

## Tilføjelse af HentUdbud SOAP service

Baseret på ovennævnte WSDL filer og tilhørende XSD-fil, defineres HentUdbud SOAP servicen til at hente hold fra HentUdbud tabellen. Her tilføjes en ny klasse, *HentUdbudWebserviceV1Impl*, som definerer og implementere servicens endpoints. Som nævnt bruges løbenummer til at understøtte at modtageren af servicen kan hente de hold der blev ændret sidst.

URL'en for dette endpoint er: */hentudbud-webservice/HentUdbudWebserviceV1*

## Request

Servicen tilgås ved at sende en SOAP request, hvori modtageren specificeres med ID, transaktions ID og et institutionsnummer. Her kan der valgfrit indsættes yderligere specificering med en liste af DS-numre samt løbenummer. Strukturen er yderligere specificeret i dokumentet for integrationsdesign: [D0180 - HentUdbud](#).

## Response

Siden afsendelse af request består af en valgfri DS-nummer liste samt valgfri specificering af løbenummer, er der fire former for svar fra servicen.

- Hvis både DS-nummer listen og løbenummer er tomme, sendes alle elementer fra HentUdbud tabellen tilbage.
- Hvis DS-nummer listen har ét eller flere elementer mens løbenummer er tom sendes alle elementer med tilsvarende institutionsnummer tilbage.
- Hvis DS-nummer listen er tom og løbenummer er sat sendes alle elementer tilbage, hvor løbenummer er større end det medsendte løbenummer

- Hvis både DS-nummer listen har ét eller flere elementer samt løbenummer er sat sendes alle elementer med tilsvarende institutionsnummer og hvor løbenummer er større end det medsendte løbenummer tilbage.

For at begrænse antallet af hold der bliver sendt tilbage i responset tilføjes der en begrænsning i form af en ny configuration property, *HENT\_UD\_BUD\_MAX\_RESULTS*, til modulet i klassen *ConfigurationEnum*. Værdien sættes i .properties filerne.

Strukturen er specificeret i dokumentet for integrationsdesign: [D0180 - HentUdbud](#).

## HentOptagedePladser

Denne sektion beskriver det detaljeret design for HentOptagedePladser webserviceen.

### Datamodel

se [D0120 - Fysisk Datamodel - HentUdbud](#)

### Definition af webservicen

På samme måde som HentUdbud webservice, defineres HentOptagedePladser også med en WSDL-fil og en tilhørende XSD fil som tilføjes til *hentudbud-webservice-interface* modulet:



### Håndtering af ændringer til tilmeldinger

Siden HentOptagedePladser skal fungere som opslag til optagede pladser på holdet for tilmeldinger, skal servicen også håndtere ændringer til perioden for tilmelding og fagtilmelding samt ændring af AktiGuid for tilmeldingen og ændringer til fag på hold for fagtilmeldingen. En oversigt over relevante ændringer ses nedenfor:

- Tilmelding
  - AktiGuid
  - Startdato
  - Slutdato
- FagTilmelding
  - Startdato
  - Slutdato
  - AmulnsituationsfagPaaHoldId

Til at håndtere disse events tilføjes endnu en observer, *HentOptagedePladserDataObserver*, som vurderer relevansen af ændringerne.

Hvis ændringen/ændringerne er relevante bliver listen af hold hentet baseret på tilmeldingen/tilmeldingernes AktiGuid, og der vil yderligere laves et nyt element for hvert hold i tabellen for HentOptagedePladser. Her gemmes data holdet hold i kolonnen *optagede\_pladser\_response* som tekst i JSON format, og kan derved hentes senere i webservicen uden at finde hold data igen.

Før et nyt element laves i HentOptagedePladser tabellen checkes der først om et element med samme AktiGuid allerede eksisterer. Hvis der allerede eksisterer et element med samme AktiGuid opdateres elementet. Hvis ikke, laves en nyt element.

Yderligere er det relevant for HentOptagedePladser at håndtere sletning af hold. Her udvides observeren til at håndtere sådanne events og slette det tilsvarende element i HentOptagedePladser tabellen.



## Dataindsamling for holdet/holdene

Siden ændringerne vedrører tilmeldinger skal det være muligt at hente data om hold baseret for AktiGuids fra tilmeldinger og fagtilmeldinger for at bygge responset. Til dette tilføjes en service, *HentOptagedePladserDataService*, i submodulet *hentudbud-service*.

HentOptagedePladser bruger samme services for Kursuskatalog som HentUdbud, altså *AmuHoldService* og *TilmeldingRestService*. Yderligere hentes skoledagskalendere for tilmelding via *TilmeldingRestService* fra Kursuskatalog.

For at hente data omkring tilmeldinger udnyttes opsætningen beskrevet for HentUdbud. Det vil sige, at der bruges samme bruger og definition af API klienter til tilmelding REST servicen.

## Beregning af FagpladsListen

For at sætte holdets fagpladser medtages dage der opfølger følgende:

- Datoen er indeholdt i en skoledag der er i skoledagskalenderen for holdet
- Datoen er indefor holdets periode
- Datoen er indeholdt i en fagperiode

Herefter beregnes hvor mange fagtilmeldinger for hvert fag der overlapper datoerne.

## Tilføjelse af HentOptagedePladser SOAP service

HentOptagedePladser SOAP service defineres ud fra ovenstående WSDL-filen og XSD-filen. En ny klasse, *HentOptagedePladserWebserviceV1Impl*, som definerer samt implementerer webservicens endpoint.

URL'en for dette endpoint er: */hentudbud-webservice/HentOptagedePladserV1*

### Request

HentOptagedePladser kan tilgås ved at sende et SOAP request, med specificeret modtager og institutionsnummer, hvori der kan sendes en valgfri liste af AktiGuids. Strukturen er specificeret i dokumentet for integrationsdesign: [D0180 - HentUdbud](#).

### Response

Webservicen sender response tilbage baseret på listen af AktiGuids. Hvis listen er tom sendes alle indgange i HentOptagedePladser tabellen tilbage. Hvis listen ikke er tom sendes kun elementer med tilsvarende AktiGuid fra tabellen tilbage. Strukturen er specificeret i dokumentet for integrationsdesign: [D0180 - HentUdbud](#).

## Initiel cache-opbygning

Ved produktionssættelse skal HentUdbud og HentOptagedePladser's persistente cache initieres, baseret på konverteret data for derefter at blive vedligeholdt via data-ændringsevents.

Til denne initiering benyttes et batchjob (*HentUdbudCachePreloadBatchlet*), som startes manuelt når Voksenuddannelse begynder at behandle konverteret data fra EU.dk.

Batchjobbet er integreret i JBoss og kan startes og monitoreres direkte i brugergrænsefladen for JBoss-administratorer.

Nedenfor ses et sekvensdiagram, som viser hvordan batchjobbet eksekverer:

