

## Opgave 1

Diskutér fordele og ulemper ved anvendelsen af XML som en fælles struktur/sprog til repræsentation af data i et Messaging System.

## Opgave 2

Hvert flyselskab har deres egen repræsentation af den information, der beskriver ankomst og afgang for selskabets fly.

Repræsentationen for de tre flyselskaber vi arbejder med i vores Case er vist herunder. Repræsentationen er ikke knyttet til et specifikt programmeringssprog.

Lad os kalde det et pseudosprog ☺

Her er de:

```
public class AirlineCompanySAS {
    String Airline;           //SAS
    String FlightNo;          //SK 239
    String Destination;        //JFK
    String Origin;             //CPH
    String ArivalDeparture     //D
    Date: dato;                //6. marts 2017
    Time: tidspunkt;           //16:45
}

public class AirlineCompanyKLM {
    String Airline;           //KLM
    String FlightNo;          //154
    String Destination;        //San Diego
    String Origin;             //Schipol
    Date/Time date/time;       //March 06 2017 16:45
}

public class AirlineCompanySW {
    String Airline;           //South West Airlines
    String FlightNo;          //SW056
    String Destination;        //New York
    Date date;                 //03/06/2017
    Time departure;            //09:45 PM
}
```

Som vi ved fra tidligere, så udveksler selskaberne information med Airport Information Center.

Lav et forslag til en Canonical Data Model for disse transaktionstyper.

### Opgave 3

Ved anvendelse af pattern(s), design og implementer en måde hvorpå transaktionerne kan transformeres til den valgte Canonical Data Model.

### Opgave 4

I en tidligere opgave definerede vi et XML-dokument, der blev genereret ved Check-In.

```
<?xml version="1.0" encoding="utf-8"?>
<FlightDetailsInfoResponse>
  <Flight number="SK937" Flightdate="20170225">
    <Origin>ARLANDA</Origin>
    <Destination>LONDON</Destination>
  </Flight>
  <Passenger>
    <ReservationNumber>CA937200305251</ReservationNumber>
    <FirstName>Anders</FirstName>
    <LastName>And</LastName>
  </Passenger>
  <Luggage>
    <Id>CA937200305252</Id>
    <Identification>1</Identification>
    <Category>Normal</Category>
    <Weight>17.3</Weight>
  </Luggage>
  <Luggage>
    <Id>CA937200305252</Id>
    <Identification>2</Identification>
    <Category>Large</Category>
    <Weight>22.7</Weight>
  </Luggage>
</FlightDetailsInfoResponse>
```

Det system der anvendes af Airport Cargo Loader Team har brug for at kende den totale vægt af hver enkelt passagers bagage samt den totale vægt for alle passagerers bagage for hvert enkelt fly.

Foreslå hvilket mønster/pattern fra [EIP], der kan anvendes til denne implementering.

### Opgave 5

Implementér mønsteret fra Opgave 4 – således, at du kan transformere den information, der udveksles mellem Check-In og Cargo Load Team.