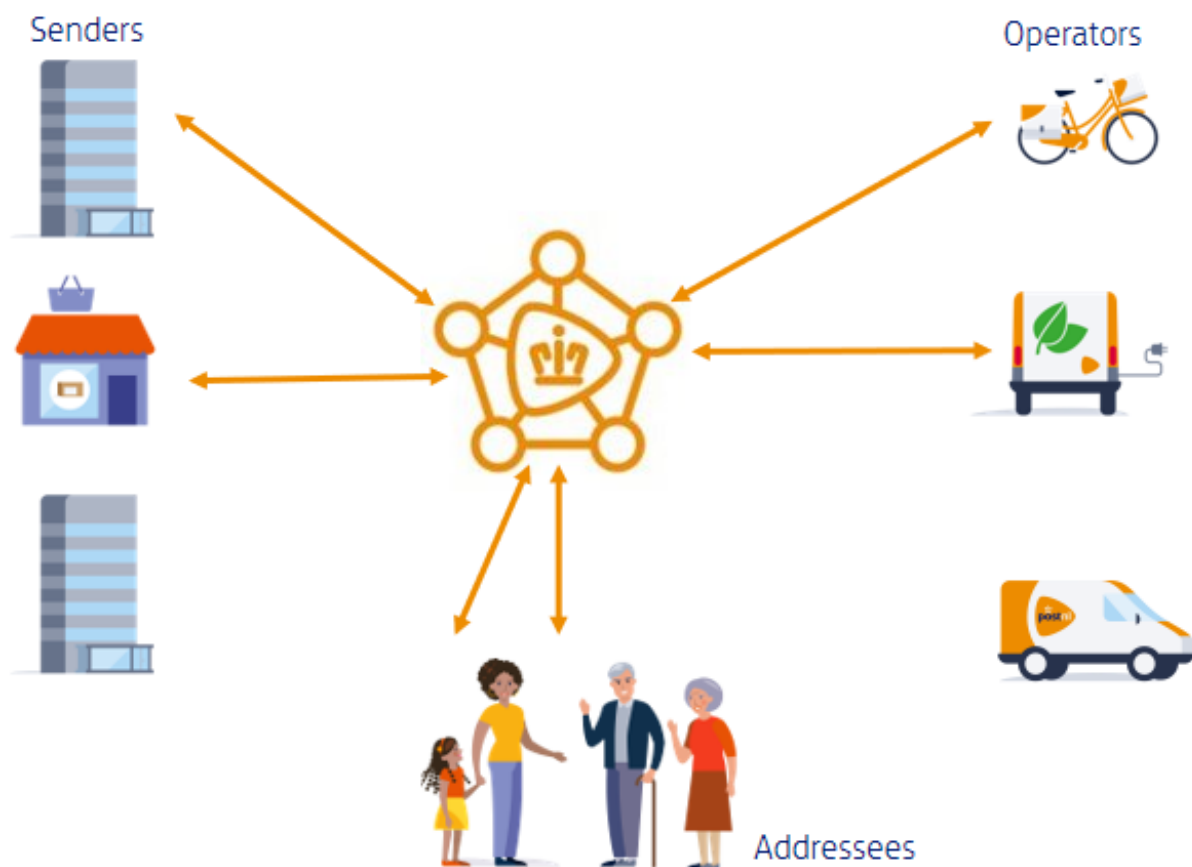


## Context

PostNL is a logistics eCommerce player and has parcel distribution within the Benelux as its core competence. To do this right, PostNL has integrations with just about all its eCommerce customers (e.g. web shops). Through these integrations these customers can do various things related to the operational process around parcel distribution. Examples are request a digital representation of a label, request Track and Trace information for a parcel on route and inform PostNL of the digital representation of the information of the parcel (pre-announcement). On the other hand, PostNL works with internal and external operators that get orders from the PostNL organization and produce events to elaborate on the status of the execution of these orders. The operator is defined as an executor on a specific part of the logistics process that PostNL executes on behalf of her customers. Operators can be business units within PostNL but can also be external partners of PostNL.



The Event Management Platform (EMP) plays a central role in the receipt, evaluation and control of the business events passing through the PostNL IT Logistics Platform that handles the information flow of PostNL's operation. It is a central component within our architecture that knows the status of each parcel and order within our operation. Upon this status, the EMP determines the ability for senders or addressees to apply various logistical actions such as re-routing or returning the parcel. This status and corresponding possibilities can be requested in real time by other systems in the landscape. The EMP also plays a crucial role in determining exceptions within our processes. It reviews all events against the execution plan and facilitates automated exception management.

The EMP is built on our Amazon AWS infrastructure using various Enterprise Open Source Components and Cloud native services. The Enterprise Open Source Components are configured/build using Javascript/Typescript. The EMP represents our central nerve system within our application landscape and is therefore characterized as business critical. If it does not work properly, customers cannot process and ship parcels with PostNL accurately. Therefore, the platform must work according to an 24x7 always-on principle and due to high flux in demand it must scale up and down with this demand without limits.

The EMP system integrates with the various applications in the application landscape of PostNL. For instance, for the synchronous communication of the status for a particular parcel or order. It also integrates with our datawarehouse environment for non-real-time storage and handling of events. It has services through which new pre-announcements and orders can be registered and it has connections with all our internal and external operators for processing their events.

### **Assignment**

Imagine our Event Management Platform as described above. It is part of our business-critical AWS infrastructure and should be designed, implemented and deployed accordingly. It must have stable performance and scale up and down with the volumes that are requested from it and must react in real-time to the dynamics of the PostNL operational organization.

You will have a 60 minute timeslot: design, present and preferably demo a solution (prototype, POC) for this platform:

1. How would you approach the design, build, test and deploy of this service AWS 'cloud natively', and 'as-a-service'?
2. What can you suggest about design patterns, software architecture and/or other measures to make the code understandable and manageable for the team?
3. What would you suggest with regard to: 24x7 business criticality, auto-scaling and auto-healing requirements?
4. What would be the outline for the Software Architecture document to be created by the team? What parts to be defined first?
5. What do you expect from a Product Owner (BizDevOps) and your engineering peers?