

## Lab 12

### Part 1

Somewhere on the interstate are 2 cameras installed. The distance between camera1 and camera2 is ½ mile. On this interstate the maximum speed is 70 miles/hour.

Both cameras make a picture of every car that passes, and the cameras have build-in image recognition software that detects and extracts the license plate. The cameras create the following record for every car that passes the camera:

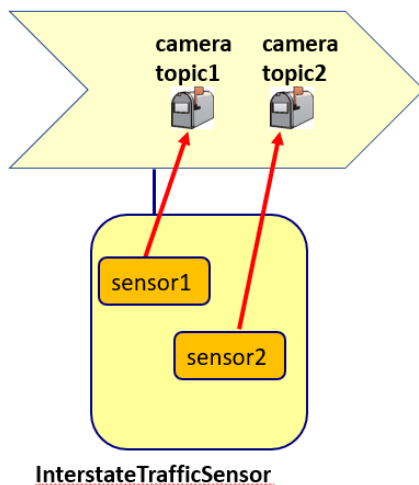
```
public class SensorRecord {  
    public String licencePlate;  
    public int minute;  
    public int second;  
    public int cameraId;  
}
```

The cameras are connected to a kafka messaging system.

Camera1 publishes its SensorRecords into the topic **cameratopic1**

Camera2 publishes its SensorRecords into the topic **cameratopic2**

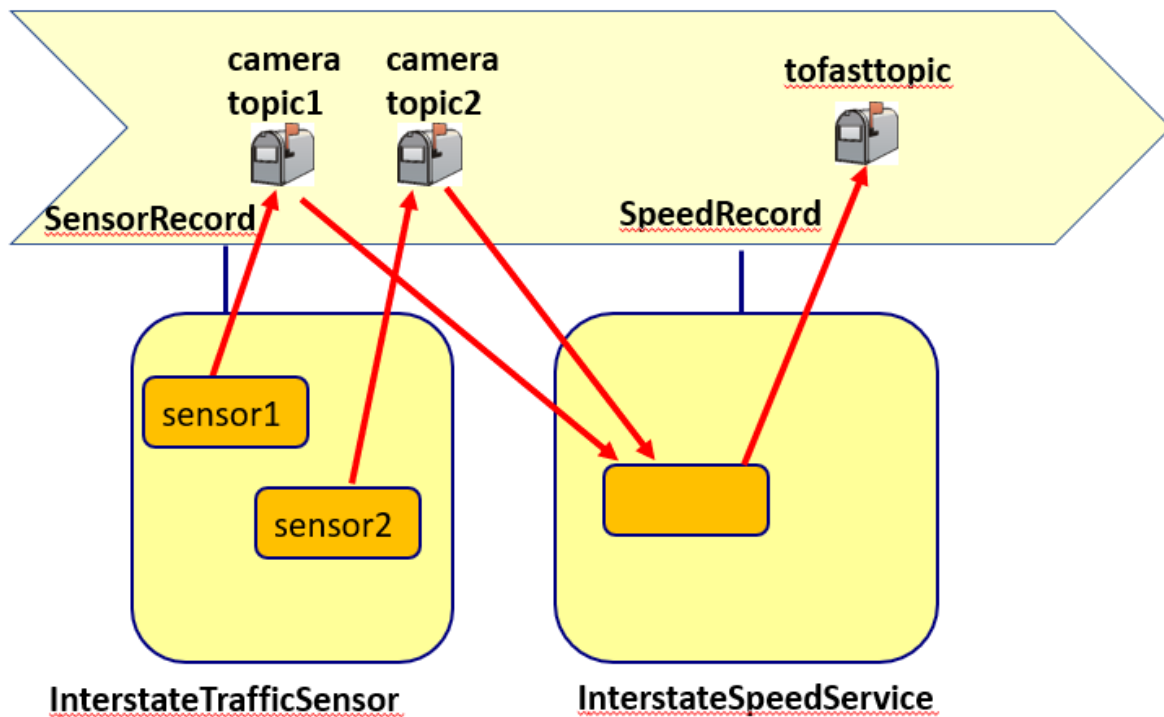
Given is the project **InterstateTrafficSensor** that simulates both cameras. If you run it, it will publish many SensorRecords into kafka.



Our first job is to calculate the speed of every car.  
(speed in miles per hour = 0.5 / time in seconds \*3600.

Then we only need to handle the cars that drive more than 72 miles/hour. Because our detection systems are not 100% accurate, we allow a maximum speed of 72 miles/hour.

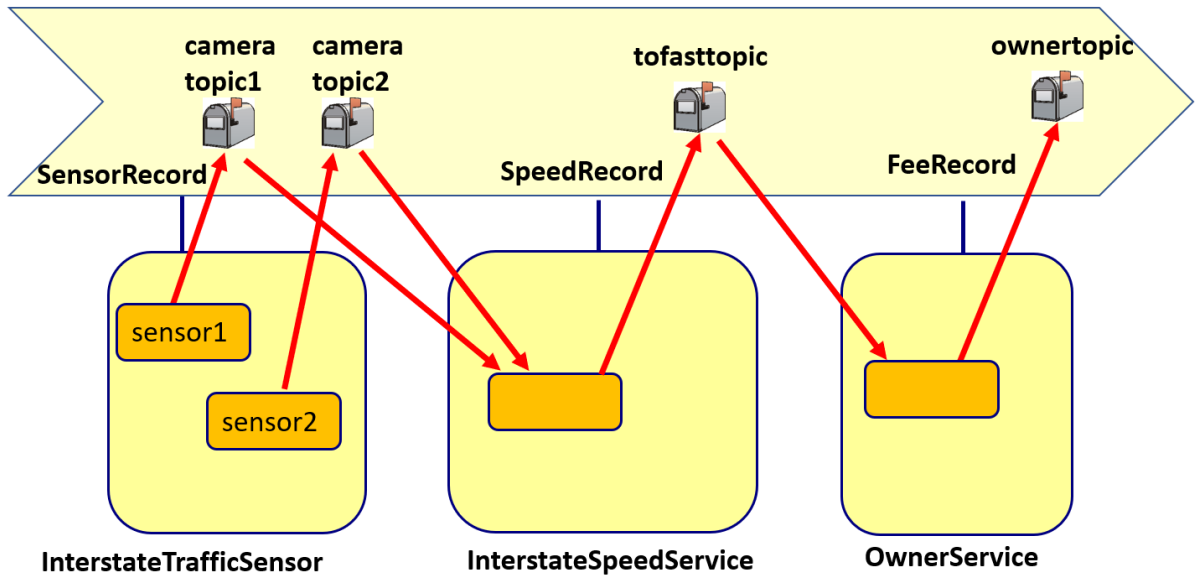
- a. Write a separate **SpeedService** that writes to the console all license plates and their corresponding speed



The algorithm to calculate the speed is given!

Once we know which cars drive more than 72 miles/hour, we need to know the owner information that is registered with the particular license plate.

- b. Write a separate **OwnerService** that writes to the console the owners info of all cars that drive more than 72 m/h all



Just make up some owner data.

If we know the owner information, then we have to calculate the fee that this owner has to pay. The formula is as follows:

72 – 77 miles/hour = \$ 25  
77 - 82 miles/hour = \$ 45  
82 – 90 miles/hour = \$ 80  
> 90 miles/hour = \$ 125

- c. Write a separate **FeeCalculatorService** that writes to the console the following information of all cars that drive too fast: license plate, owner info, speed, amount of the fee

## **Part 2**

# Financial Risk System

### **Background**

A global investment bank based in London, New York and Singapore trades (buys and sells) financial products with other banks (counterparties). When share prices on the stock markets move up or down, the bank either makes money or loses it. At the end of the working day, the bank needs to gain a view of how much risk they are exposed to (e.g. of losing money) by running some calculations on the data held about their trades. The bank has an existing Trade Data System (TDS) and Reference Data System (RDS) but need a new Risk System.

### **Trade Data System**

The Trade Data System maintains a store of all trades made by the bank. It is already configured to generate a file-based XML export of trade data at the close of business (5pm) in New York. The export includes the following information for every trade made by the bank:

- Trade ID
- Date
- Current trade value in US dollars
- Counterparty ID

### **Reference Data System**

The Reference Data System maintains all of the reference data needed by the bank. This includes information about counterparties; each of which represents an individual, a bank, etc. A filebased XML export is also available and includes basic information about each counterparty. A new organization-wide reference data system is due for completion in the next 3 months, with the current system eventually being decommissioned.

### **Functional Requirements**

The high-level functional requirements for the new Risk System are as follows.

1. Import trade data from the Trade Data System.
2. Import counterparty data from the Reference Data System.
3. Join the two sets of data together, enriching the trade data with information about the counterparty.
4. For each counterparty, calculate the risk that the bank is exposed to.
5. Generate a report that can be imported into Microsoft Excel containing the risk figures for all counterparties known by the bank.
6. Distribute the report to the business users before the start of the next trading day (9am) in Singapore.
7. Provide a way for a subset of the business users to configure and maintain the external parameters used by the risk calculations.

## **Non-functional Requirements**

The non-functional requirements for the new Risk System are as follows.

### **Performance**

- Risk reports must be generated before 9am the following business day in Singapore.

### **Scalability**

- The system must be able to cope with trade volumes for the next 5 years.
- The Trade Data System export includes approximately 5000 trades now and it is anticipated that there will be an additional 10 trades per day.
- The Reference Data System counterparty export includes approximately 20,000 counterparties and growth will be negligible.
- There are 40-50 business users around the world that need access to the report.

### **Security**

- This system must follow bank policy that states system access is restricted to authenticated and authorized users only.
- Reports must only be distributed to authorized users.
- Only a subset of the authorized users are permitted to modify the parameters used in the risk calculations.

### **Audit**

- The following events must be recorded in the system audit logs:
  - Report generation.
  - Modification of risk calculation parameters.
- It must be possible to understand the input data that was used in calculating risk.

### **Fault Tolerance and Resilience**

- The system should take appropriate steps to recover from an error if possible, but all errors should be logged.
- Errors preventing a counterparty risk calculation being completed should be logged and the process should continue.

### **Monitoring and Management**

- A Simple Network Management Protocol (SNMP) trap should be sent to the bank's Central Monitoring Service in the following circumstances:
  - When there is a fatal error with a system component.
  - When reports have not been generated before 9am Singapore time.

### **Data Retention and Archiving**

- Input files used in the risk calculation process must be retained for 1 year.

- a. Draw the use case diagram
- b. Draw the activity diagram that shows how to create a risk report
- c. Draw the Context diagram
- d. Draw the Container diagram
- e. Draw the component diagram
- f. Draw a sequence diagram that shows how the components work together to create a risk report.