Exercises from some written exams

Availability assessment

- Assume you have available three components offering the same functionality with different availability:
  - LOO shows 95% availability
  - BOO shows 98% availability
  - MOO shows 70% availability
- What component configuration (combination of the three components) would you use to obtain an overall availability equal to or above 99%?
  - Note that you can use only one copy of each component, and you could use fewer than all available components.

Availability assessment - solution

- the most natural way to achieve >=99% availability is to place components in parallel.
- It turns out it is sufficient to place BOO in parallel to LOO to achieve: 1-(0.02*0.05)=0.999 availability.
- If LOO could be available in multiple instances, even two instances of LOO would be sufficient: 1-(0.05*0.05)=0.9975

Exercise 1 on architecture definition

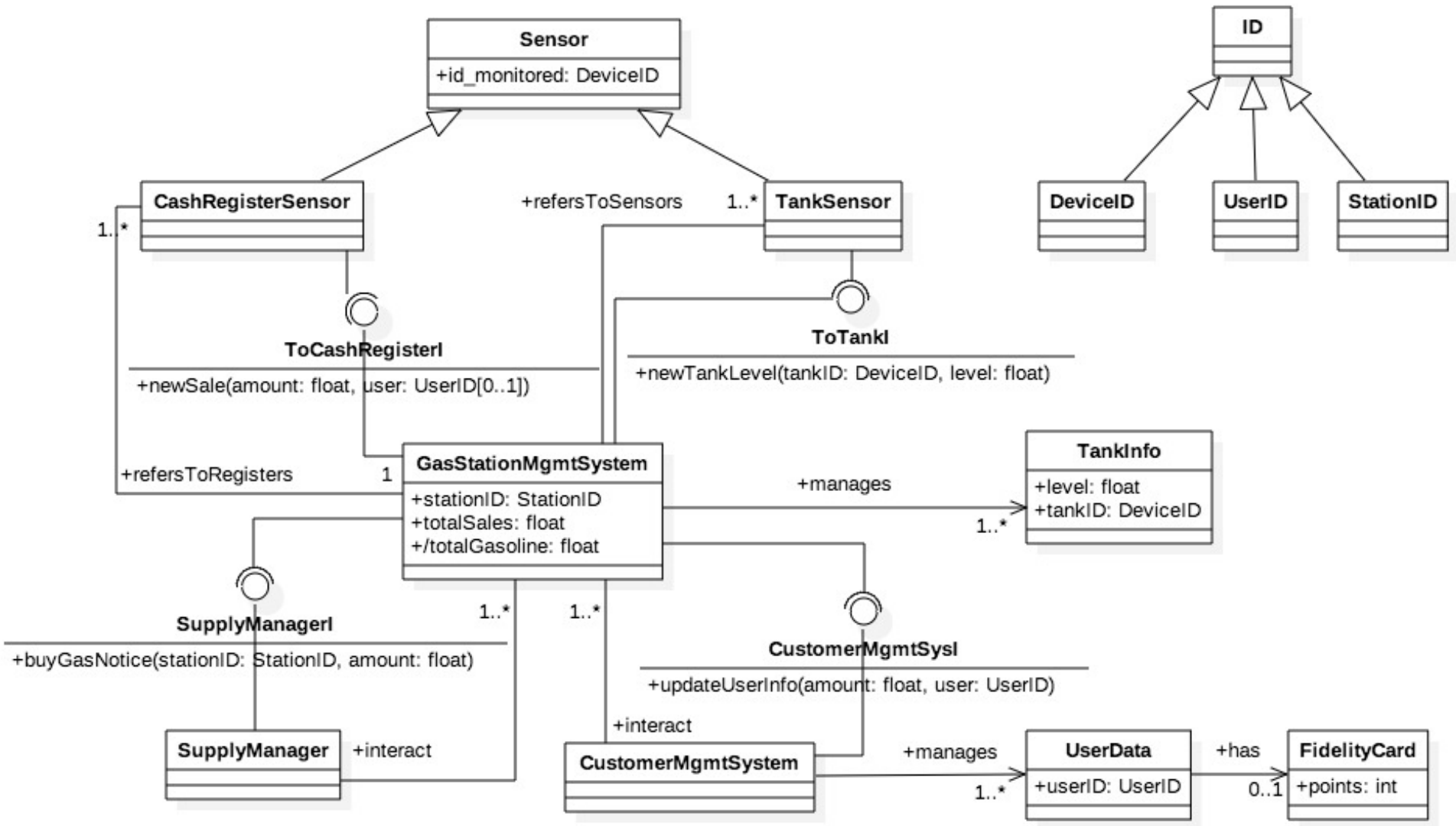
- A company managing gas stations aims at developing an information system offering the following functions:
  1. Visualize the status of all gas stations in terms of quantity of gasoline available and sales. The main system acquires data from sensors located within the gasoline tanks and from the cash registers.
  2. Manage the process through which a customer refuels his/her tank. If the customer has a fidelity card, the points corresponding to the gasoline purchase have to be added to this card.
  3. Alert the supplying department about the need to buy new gasoline when it goes below a certain threshold.
  4. Manage the fidelity cards assigned to customers, e.g., to assign points, apply discounts, …

Exercise 1 on architecture definition (cont.)

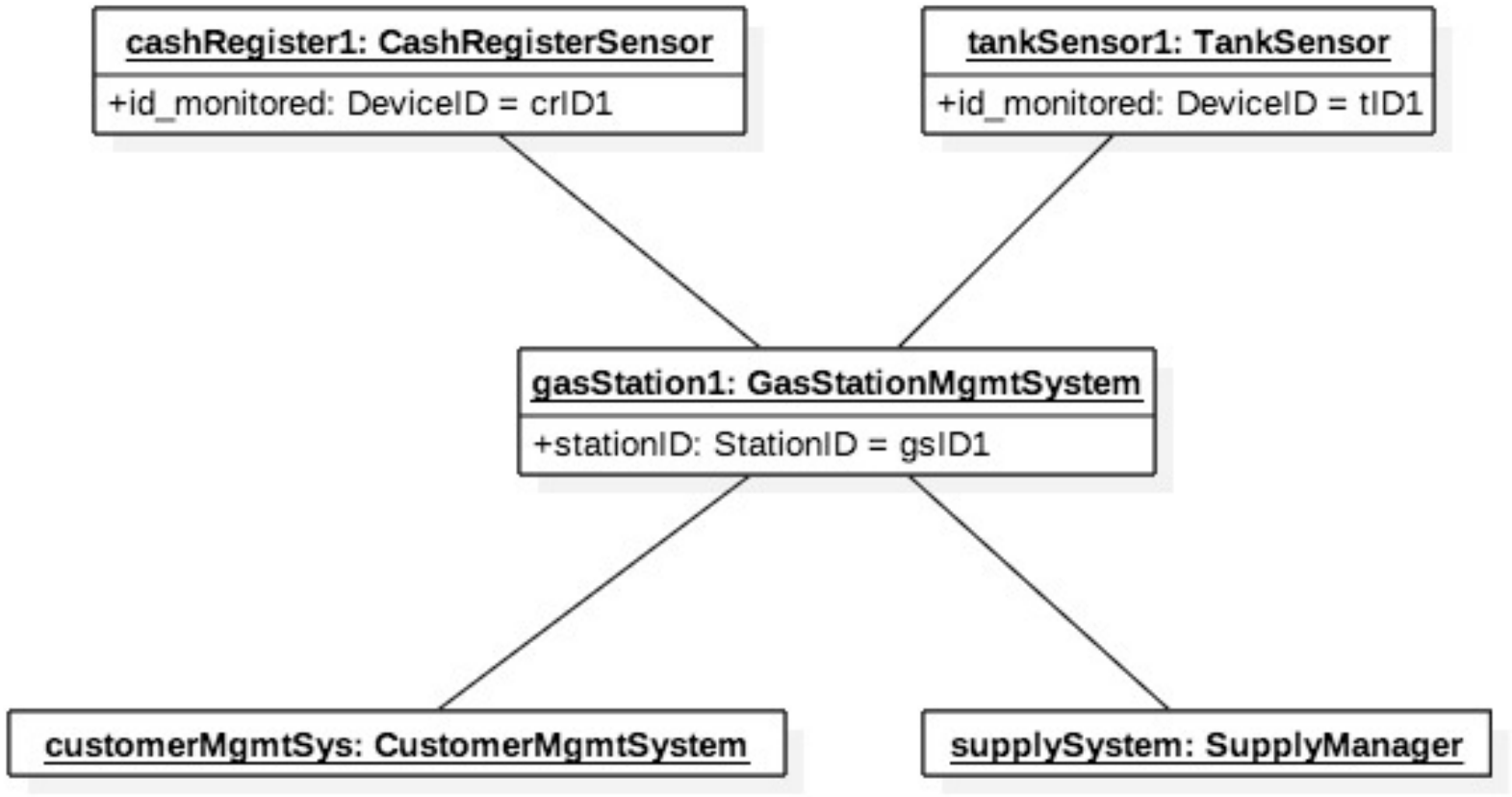
- Define the system architecture.
  - List all components belonging to the architecture, describe their functions and the way they interact with each other.
  - Draw a UML class or component diagram to provide a visual representation of the architecture.
  - Define a UML sequence diagram that describes the interaction between components needed to accomplish function 3 (alert the supplying department about the need to buy new gasoline).

Solution: class diagram

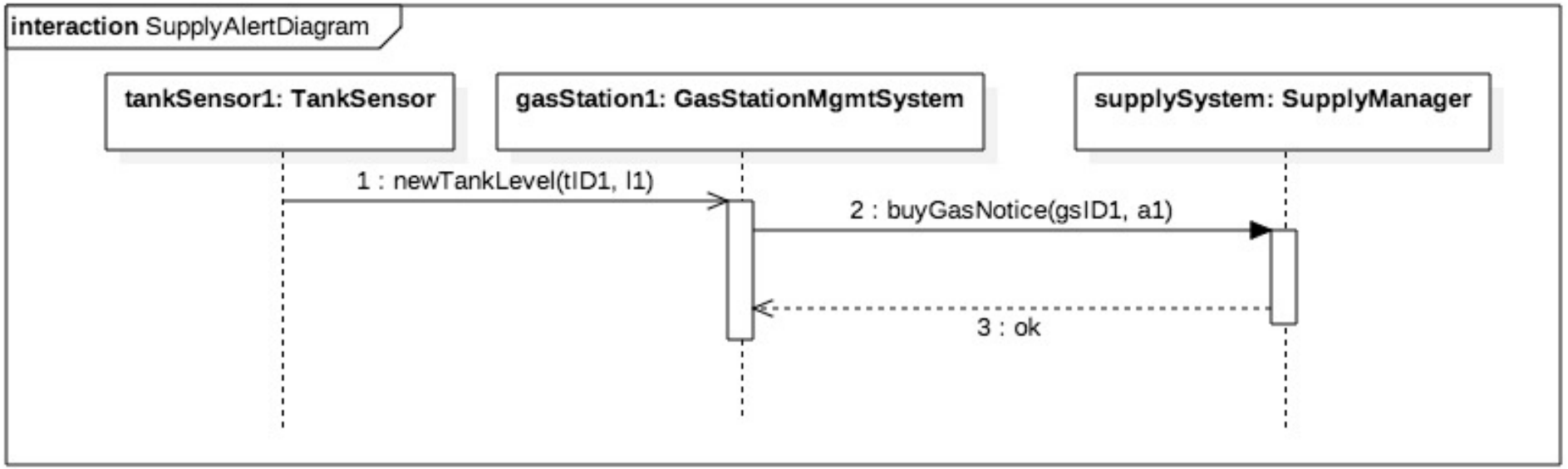
(without description; note that the exercise requires you to include also a description of elements)



Solution: object diagram
(not required, but useful in this case)



Solution: sequence diagram



Exercise 2 on architecture definition

- A public transportation company of a large urban area aims at opening a service for enabling sharing of electric vehicles. The vehicles can be cars, scooters and electric bicycles. The company offers to its users a number of vehicles and the charging stations plugs. Each user can either rent a vehicle (only one at a time), or use the charging stations provided by the company to recharge his own vehicle.
- You have been contacted by the company to develop the software system supporting the operation of the service.

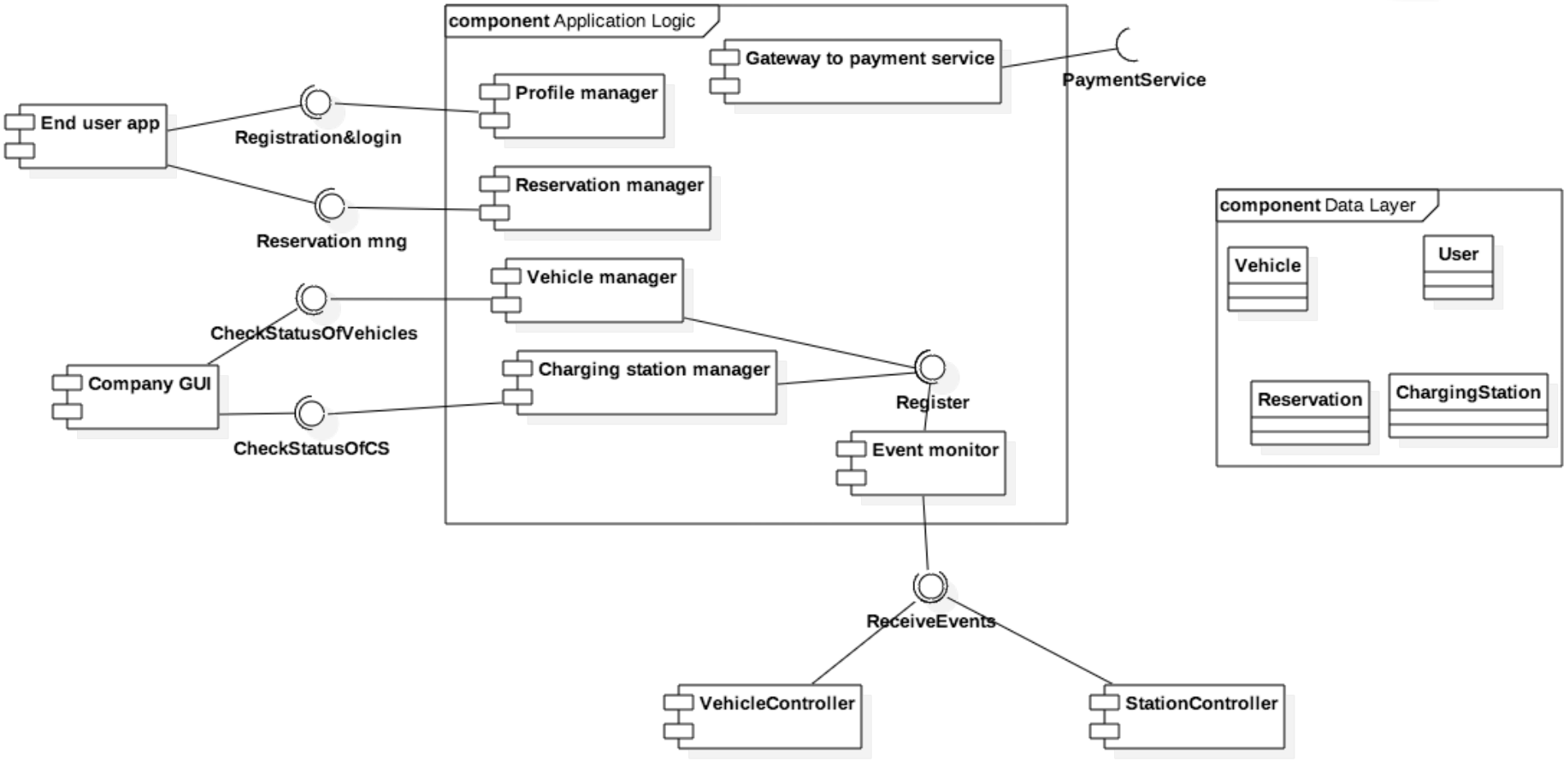
Exercise 2 on architecture definition (cont.)

- This system should allow:
  - Customers to place/modify/remove their reservations. When a customer places a reservation he/she should specify the time slot in which he/she will use the vehicle.
  - Customers to return a vehicle or a charging station plug. Vehicles have to be returned before the time slot associated to the corresponding reservation expires.
  - The company to enter information about vehicles and charging stations, to monitor the actual situation of its vehicles and charging stations, to charge users for the usage of their service.
- From the side of customers, the service should be accessible through a mobile device.

Exercise 2 on architecture definition (cont.)

- Define the high level architecture of the software system. Describe each component and the way it is connected to the others.
- Define the UML sequence diagrams to describe the behaviour of the system in the following case:
  - A customer is using one of the vehicles of the company and the time slot he has declared at the reservation is expiring. Thus, the system sends to the customer information about the closest charging stations where he can leave the vehicle. The customer replies asking for charging stations available in a different area. The system provides them together with the information on the extra-time needed to reach these other charging stations (this extra-time is calculated by relying on an external map service).

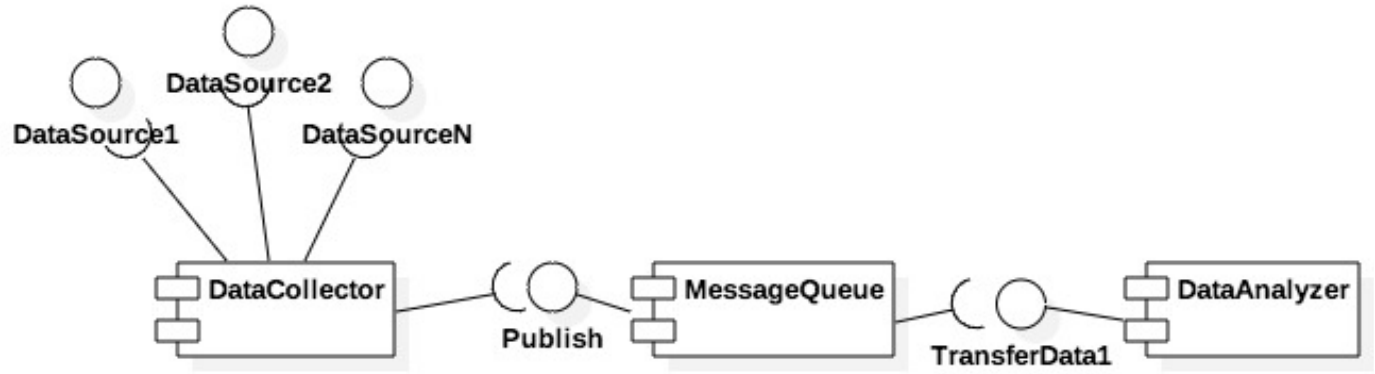
Incomplete high level architecture



- You are asked to complete it as needed

Exercise on Alloy and architectures
(exam of July 13th, 2018)

- Consider the following UML component diagram.



This diagram describes a software system that acquires and elaborates information from a number of different sources by polling them periodically. The `DataCollector` component is exploiting the interfaces offered by some data sources to acquire data and the interface of the `MessageQueue` to pass collected data to the other components. The `MessageQueue` exploits the interface offered by the `DataAnalyzer` to pass the data to this component.

Exercise (cont.)

- Write in Alloy the signatures that model a `DataSource`, a `DataCollector`, a `MessageQueue` and a `DataAnalyzer`. Make sure that you represent in the model the connections between components that are highlighted in the UML component diagram.
- Assume that we decide to replicate the `DataCollector` component. Model in Alloy the following possible configurations of the system:
  - **Configuration 1**: Each `DataCollector` replica is connected to a disjoint subset of `DataSource` components.
  - **Configuration 2**: All `DataCollector` replicas are connected to all `DataSource` components.
  - **Configuration 3**: `DataCollector` components are classified in master and slaves. There is always one `DataCollector` that acts as *master*.

A possible solution

```
sig DataSource {}
sig DataCollector {
  sources: set DataSource,
  queue : MessageQueue
}

sig MessageQueue {
  analyzer: DataAnalyzer
}

sig DataAnalyzer {}
```

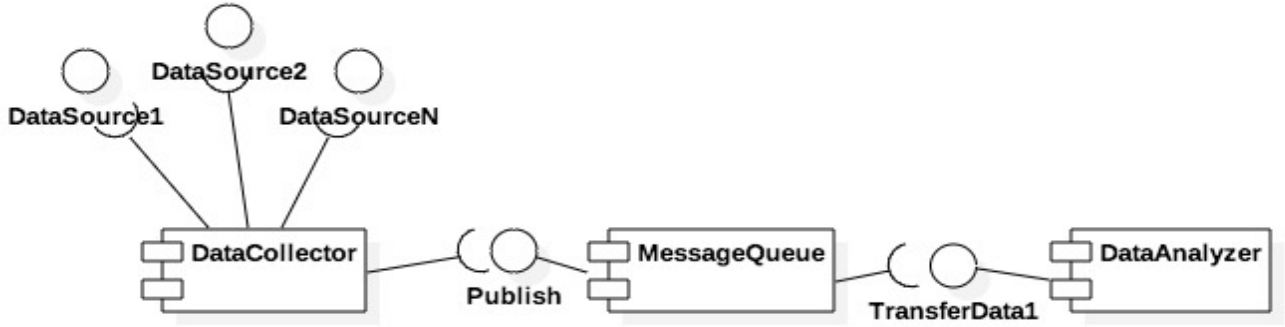
A possible solution (cont.)


```
sig Configuration {
  sources: set DataSource,
  collectors: set DataCollector,
  queue: MessageQueue,
  analyzer: DataAnalyzer
}
{ // all DataCollector components are connected to the
  // same MessageQueue, which is connected to the
  // DataAnalyzer of the configuration
  all coll : collectors | coll.queue = queue
  queue.analyzer = analyzer
  // also, the DataSource components used by the
  // DataCollector ones are exactly
  // those of the configuration
  collectors.sources = sources
}
```

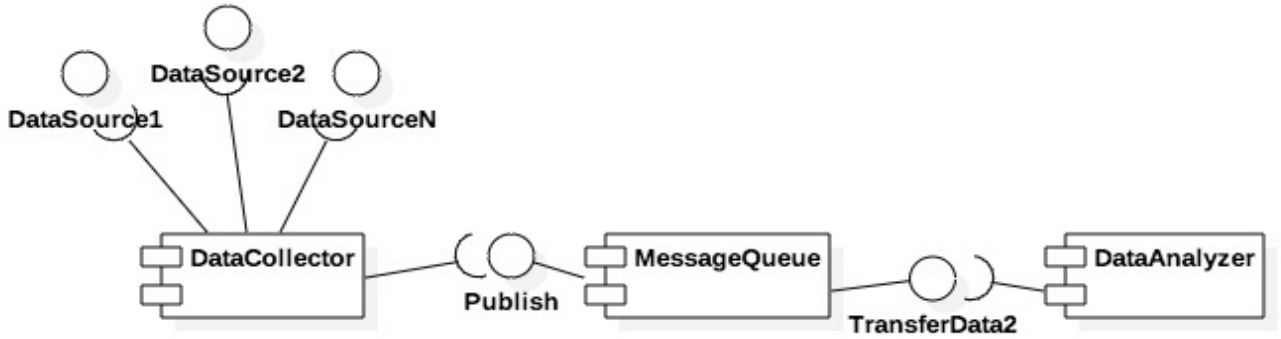
A possible solution (cont.)

```
// We capture the different configurations through
// extensions of the Configuration
// signature above; they add the necessary constraints
sig Configuration1 extends Configuration{}
{ all disj coll1, coll2 : collectors |
           coll1.sources & coll2.sources = none }

sig Configuration2 extends Configuration{}
{ all coll : collectors | coll.sources = sources }

sig MasterDataCollector extends DataCollector {}
sig SlaveDataCollector extends DataCollector {}

sig Configuration3 extends Configuration{}
{ all coll : collectors |
    coll in (MasterDataCollector | SlaveDataCollector)
  one coll : collectors | coll in MasterDataCollector
}
```

More on the data analysis example
(from the exam of June, 27th 2018)



- Consider this second version of the same system



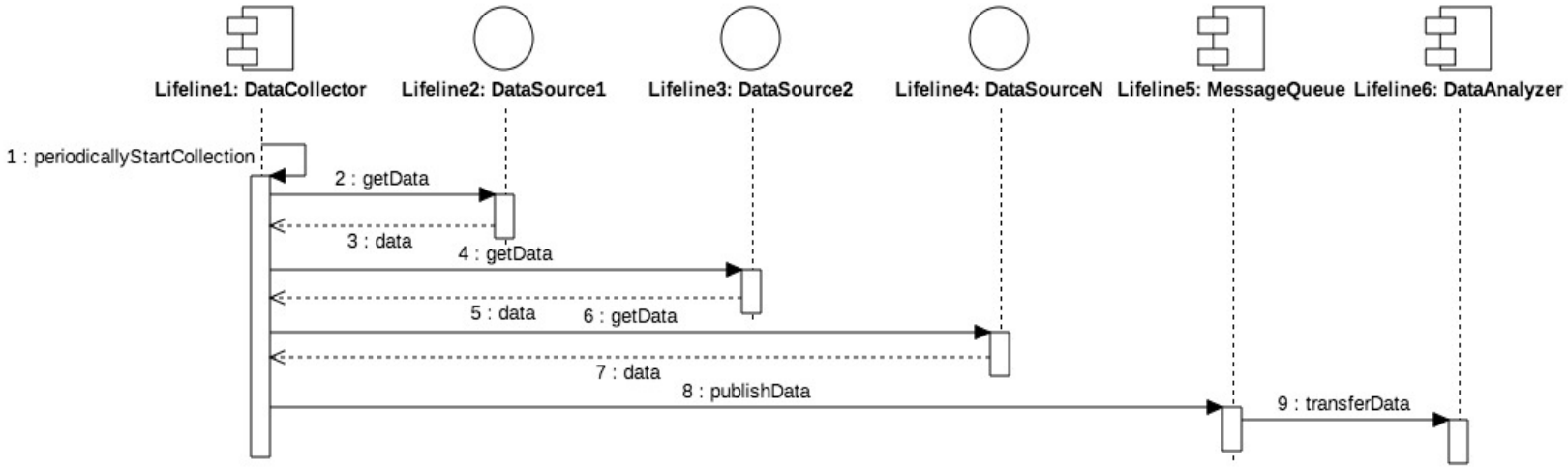
- Q1: What is the difference between the two?

More on the data analysis example
(from the exam of June, 27th 2018)

- Solution
  - In the second case, the MessageQueue does not actively push the data to the DataAnalyzer, but it offers interface TransferData2 so that the DataAnalyzer can pull data as soon as it is ready to process them. Also in this case, both a batch or a per data approach is possible. The rest of the system behaves as first one.
- Q2: Define two sequence diagrams that describe how data flow through the system in the two versions of the architecture

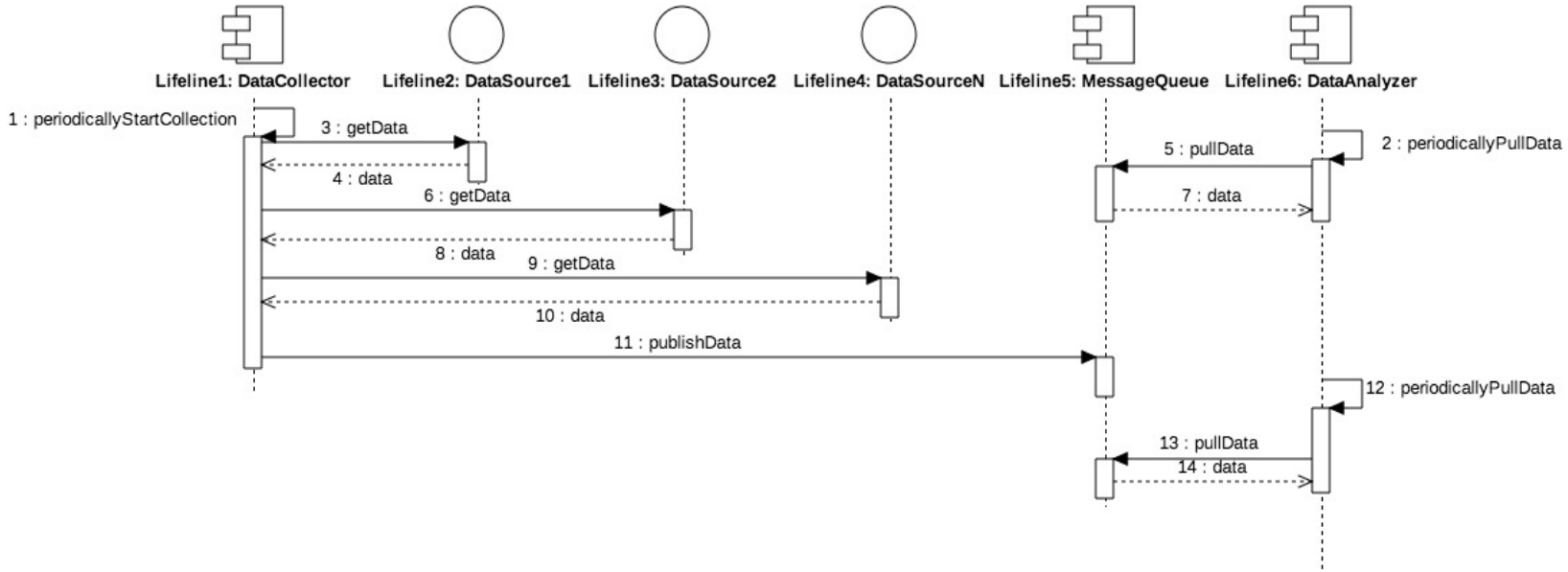
More on the data analysis example
(from the exam of June, 27th 2018)

- Solution to Q2
  - Sequence diagram compatible with the first component diagram



More on the data analysis example
(from the exam of June, 27th 2018)

- Solution to Q2
  - Sequence diagram compatible with the second component diagram



More on the data analysis example
(from the exam of June, 27th 2018)

- Assume that the components of your system offer the following availability:
  - DataCollector: 99%
  - MessageQueue: 99.99%
  - DataAnalyzer: 99.5%
- Provide an estimation of the total availability of your system (you can provide a raw estimation of the availability without computing it completely).

More on the data analysis example
(from the exam of June, 27th 2018)

- Data flow through the whole chain of components to be processed => series of component.
- The total availability of the system is determined by the weakest element, that is, the DataCollector.
  - A_{Total} = 0.99*0.9999*0.995 = 0.985
- Assuming that you wanted to improve this total availability by exploiting replication, which component(s) would you replicate? Please provide an argument for your answer.

More on the data analysis example
(from the exam of June, 27th 2018)

- If we parallelize the data collector adding a new replica, we can achieve the following availability:
  - $(1-(1-0.99)^2) * 0.9999 * 0.995 = 0.995$
- if we increase the number of DataCollector replica, we do not achieve an improvement as the weakest component becomes the DataAnalyzer.
- We can parallelize this component as well to further improve the availability of our system.

- How would such replication impact on the way the system works and is designed?