

# Report: Distributed Cloud Applications

de Grefte Sterre (*r0763644*)

Verhaeren Morganne (*r0713641*)

November 25, 2022

**Question 1** The backend of our project is composed of the SecurityFilter.java class and all the class in the entities package. The frontend exists of the ViewController.java and the Model.java classes. The backend can be distributed over hosts. The diagram is shown in figure .

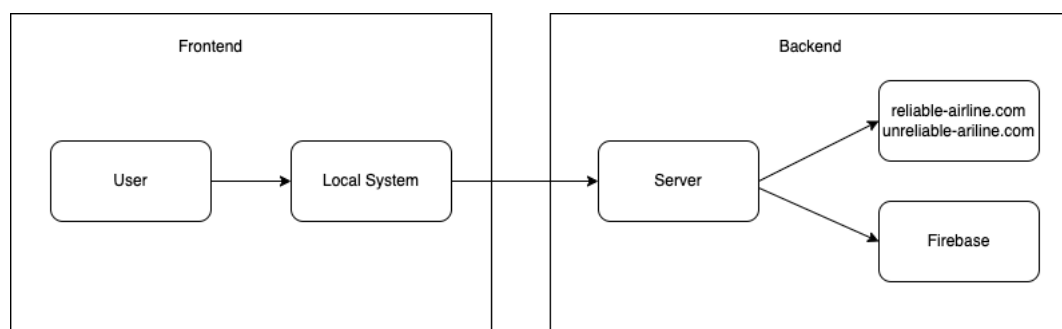
**Question 2** We could see Spring Boot as a form of middleware to hide complexity. The Firebase Authentication is another form of middleware we use to hide complexity but also to speed up the development.

**Question 3** The indirect communication between objects kicks in when trying to create a booking. The indirect communication we use here is publisher-subscriber. At the moment the user clicks the button "book all" in the cart, the application will create a push message to create the booking and update the tickets.

**Question 4** The data passed between the application and the background worker when creating a booking is all the data of the booking itself. This contains an id, a time, the tickets, and a customer. This is passed using Cloud Pub/Sub, and since this guarantees at-least-once delivery, there is no reason to persist data and send only references to this data.

**Question 5** As solution for indirect communication we use the Cloud Pub/Sub, and this improves scalability since many others servers can get information about the bookings by simply subscribing to the topic. The responsiveness is also a lot better with this Pub/Sub solution. This is because the application is idle while the requests are being handled by the background worker, meaning users can still make requests.

**Question 6** PubSub would not be beneficial if a user would try to book multiple tickets but would put them in different bookings. This would lead to multiple small messages being published, which would create some overhead and not be very beneficial.



**Question 7** No, if one of the seats in a booking is already reserved, the entire booking will be cancelled. All the seats that were already reserved will be freed again.

**Question 8** Because of role-based access control we only need to check one thing, if the user is a manager or not, to show the manager page. This is a very profitable way of authentication in distributed systems because the manager methods could be spread across multiple servers. With the use of this JWT token, every server could easily be certain that this user is authenticated and allowed to use the manager methods.

**Question 9** If we would switch to another authentication provider, we would only have to make changes in the *SecurityFilter.java* class. But since the JWT token can provide roles to certain users, we will have to make sure that this other authentication provider also could provide roles. Otherwise, a lot of the code would have to change, since a lot is based upon the role system.

**Question 10 Morganne** Our application copes with failures of the Unreliable Airline by retrying request if they fail. The application will retry each request a maximum of 3 times. If the Unreliable Airline fails more than 3 times our application will crash and users will have to try again.