Distributed Systems (H0N08A)

Report: Project

Floris Ingelaere *(r0832373)*
Karam Khalil *(r0736813)*

November 17, 2023

1. Imagine you were to deploy your application to a real cloud environment, so not a lab deployment where everything runs on the same machine. Which hosts/systems would then execute which processes, i.e., how are the remote objects distributed over hosts? Clearly outline which parts belong to the front and back end, and annotate with relevant services. Create a component/deployment diagram to illustrate this: highlight where the client(s) and server(s) are.

The Diagram is at the end of the report.

2. Where in your application were you able to leverage middleware to hide complexity and speed up development?

We were able to use middleware in the booking part. We used message passing in the form of google pub/sub and used access managing in the form of firebase authentication.

3. At which step of the booking workflow (create quote, collect cart, create booking) would the indirect communication between objects or components kick in? Describe the steps that trigger the indirect communication, and what happens afterwards.

Indirect communication kicks in during the create booking process. When trying to book the tickets, the apiController sends the order through the "bookings" topic. The push subscription receives this order from the topic and sends a signal to the repository that it needs to book that order. Afterwards the tickets get booked.

4. Which kind of data is passed between the application and the background worker when creating a booking? Does it make sense to persist data and only pass references to that data?

The list of quotes is passed between the application and the background worker. It would make no sense to use pass references to the data because different users ordering the same tickets would cause the system to think it's 1 order.

5. How does your solution to indirect communication improve scalability and responsiveness (especially in a distributed/cloud context)?

By having the booking process be dependant on pub/sub it is possible to have multiple servers provide the booking feature. This is needed for scalibility because it is a costly operation that may be cumbersome if all done on the server. It's also more responsive because the the servers can do something else while waiting for the message.

6. Can you give an example of a user action where indirect communication would not be beneficial in terms of the trade-off between scalability and usability?

And example of a user action where indirect communication wouldn't be needed is getTrains, because you can already get a simple, quick and direct access point to retrieve train data. If you use indirect communication you would just pass the message through an additional point without any benefit and only more delay.

7. Is there a scenario in which your implementation of ACID properties may lead to double bookings of one seat?

I think it is impossible, because the server counts on the PUT response from the train company to make the booking. If the seat is not reserved which means that the PUT response a 404 not found then the booking is not valid and not registered and all seats that were reserved must be released so double booking on our side can never happen it depends of how the train company handles the PUT request when multiple client do the same request.

8. How does role-based access control simplify the requirement that only authorized users can access manager methods?

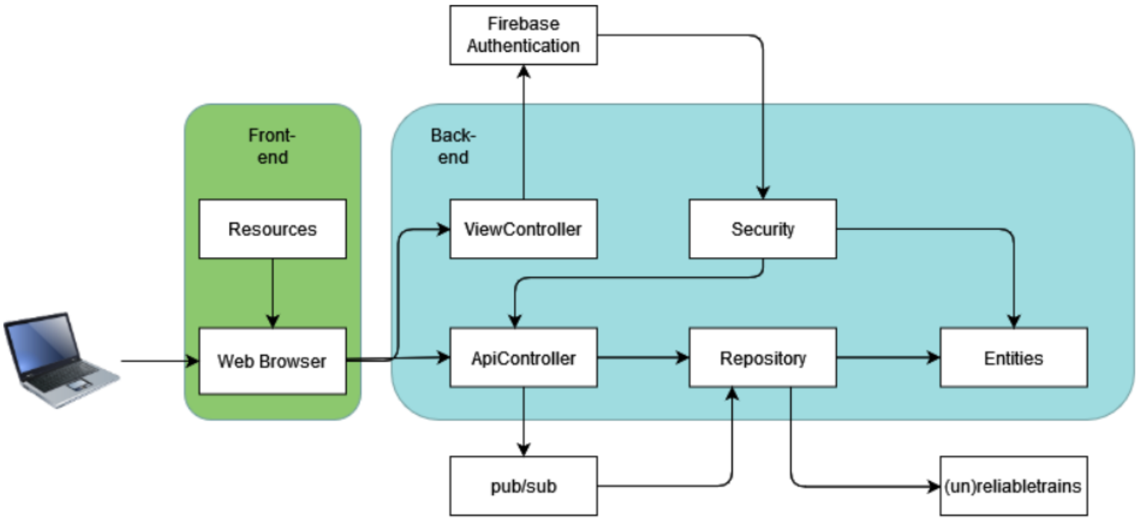
The role-based access control simplifies it by having a build in function to check whether the logged in user has the required roles to call for the methods that require authorized access.

9. Which components would need to change if you switch to another authentication provider? Where may such a change make it more/less difficult to correctly enforce access control rules, and what would an authentication provider therefore ideally provide?

We would need to change the authentication provider in how you need to assign the roles properly to the user class, the authenticator would ideally provide the roles in form of a string so they could be assigned easily as in the current implementation.

10. How does your application cope with failures of the Unreliable Train Company? How severely faulty may that train company become before there is a significant impact on the functionality of your application?

The application handles the error of the unreliable train by returning an empty object if an error occurs. In general these errors have not so much effect on the app except when trying to book too much tickets from the unreliable company. Because booking a seat happens individually the larger the number of seats that you are trying to book the larger the chance that the unreliable company gives an error and because of the ACID property you can't book any ticket.


Firebase Authentication
Front-end
Back-end
Resources
ViewController
Security
Web Browser
ApiController
Repository
Entities
pub/sub
(un)reliabletrains