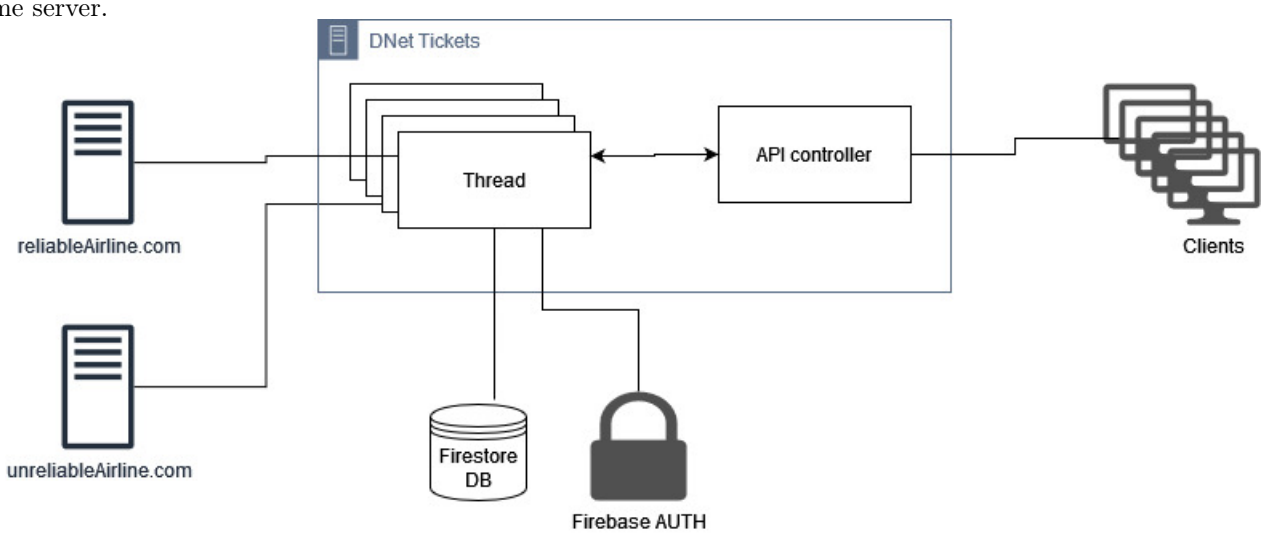
Distributed Systems (H0N08A)

Report: Project

Nicolas Scheers *(r0868851)*
Brian Verbanck *(r0904532)*

November 25, 2022

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 DNet-Ticket system consists of a webpage serving part providing the clients with a web app, and a back-end part handling the business end like fetching flights from external airlines, persisting data to the FireStore DB and handling authentication with FireBase Auth. With multiple threads, the back-end can work asynchronously for some tasks, and through pub-sub, these threads don't even have to be on the same server.



2. Where in your application were you able to leverage middleware to hide complexity and speed up development? Spring boot allows for fast creation of REST api's. The use of firebase authentication with JWT is also very straightforward. These are systems we didn't have to create ourselves but could just use by implementing a simple library or API.

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. We didn't have enough time to finish this in the code, but we would do it as follows: When we want to fetch all available flights from multiple airlines, we could leverage indirect communication to fetch all of them at the same time. Also, when the createQuote endpoints get called, We get Quotes from the airplane system which we use to create tickets, reserve seats, and create bookings. The

tickets are stored in our application while the seat's get reserved with a reference to our booking. In case of multiple tickets per booking, we could reserve those concurrently.

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? We pass a bookingReference with each ticket we reserve. A reference should be enough information for the airports, since they don't need to know anything about the bookings themselves. And if they do it's easy to write a GET request to get more information about the booking using the bookingReference. In this manner we also prevent the airport from holding unnecessary amounts of data.

5. How does your solution to indirect communication improve scalability and responsiveness (especially in a distributed/cloud context)? Our indirect communication allows airports to scale their system without breaking ours. Also we can scale ours without breaking theirs. By using the airports systems we didn't have to create our own logic for it, thus increasing the easiness of creating a working system.

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? Retrieving one's bookings. It would not be beneficial to use indirect communication here since the booking functionality is our own.

7. Is there a scenario in which your implementation of ACID properties may lead to double bookings of one seat? No following the ACID properties this shouldn't be possible.

8. How does role-based access control simplify the requirement that only authorized users can access manager methods? Al requests send their JWT, this makes users able to use their designated endpoints without needing additional security. (Since the JWT has the roles in it)

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? In the security filter it would become more difficult to get the authentication/control rules. Another authentication provider would ideally provide us with an authentication token in the header like a JWT does, and ideally we should be able to get all the info we need from it.

10. How does your application cope with failures of the Unreliable Airline? How severely faulty may that airline become before there is a significant impact on the functionality of your application? We solved it easily by adding retry's to our requests. The airline may fault 3 times before the app wouldn't work properly. A simple refresh should solve it further.