AeroTelcel

Marcos Gonzalez, Jorge Ramirez,
Javier Ortega

School of Computing and Communications
Lancaster University Leipzig, Germany

m.gonzalezfernandez@lancaster.ac.uk | j.ramirez1@lancaster.ac.uk |
j.ortegamendoza@lancaster.ac.uk

Agenda

- Roles in the project
- Software Requirements
- Software Design and Architecture
- Software Implementation
- Software Testing

Link to the Github Page of the project:

Roles in the Project

Lancaster University Leipzig

Student Name	Requirements	Design	Implementation	Testing
Marcos Gonzalez Fernandez	• Front-end Technology Selection	• Graphics User Interface Design • UML Class Diagram	• Graphics User Interface. • Map API.	• Unit Test
Jorge Ramirez de Diego	• Non-Functional Requirements- • Functional Requirements • UML Use Case Diagram • Back-end & Database Technology Selection	• Architectural Design • System Flow Diagram • UML Component Diagram	• SubscriptionRedis • SubscriptionHandler • Gradle Build Scripts • Dockerization of Microservices	• Subscription Lifecycle Integration Test
Javier Ortega Mendoza	• AviationData • Flight, Airport and Full Flight Object Database handling	• ER Diagrams	• Dockerization of Microservices • Flight Data API data retrieval, data sanitation, airline assignation and validation	• Unit Test - JSONification of flight objects

Lancaster University Leipzig

Software Requirements

Functional Requirements

Lancaster University Leipzig

- **Flights and Airports**
- The system shall provide real-time information of flights and airports.
- The system shall be capable of letting users search based on flight number or code.
- The system shall be capable of letting users search for a flight based on the airport code departure and arrival.
- The system shall show the map with the coordinates of airports and flights.

- **Subscriptions and Notifications**
- The system shall allow the user to subscribe to receive notifications from a particular flight or airport.
- The system shall allow the user to unsubscribe to stop receiving notifications from a particular flight or airport.

- **User Feedback/Reports**
- The system shall allow users to rate specific flights or airports based on various qualities.

Non-Functional Requirements

Lancaster University Leipzig

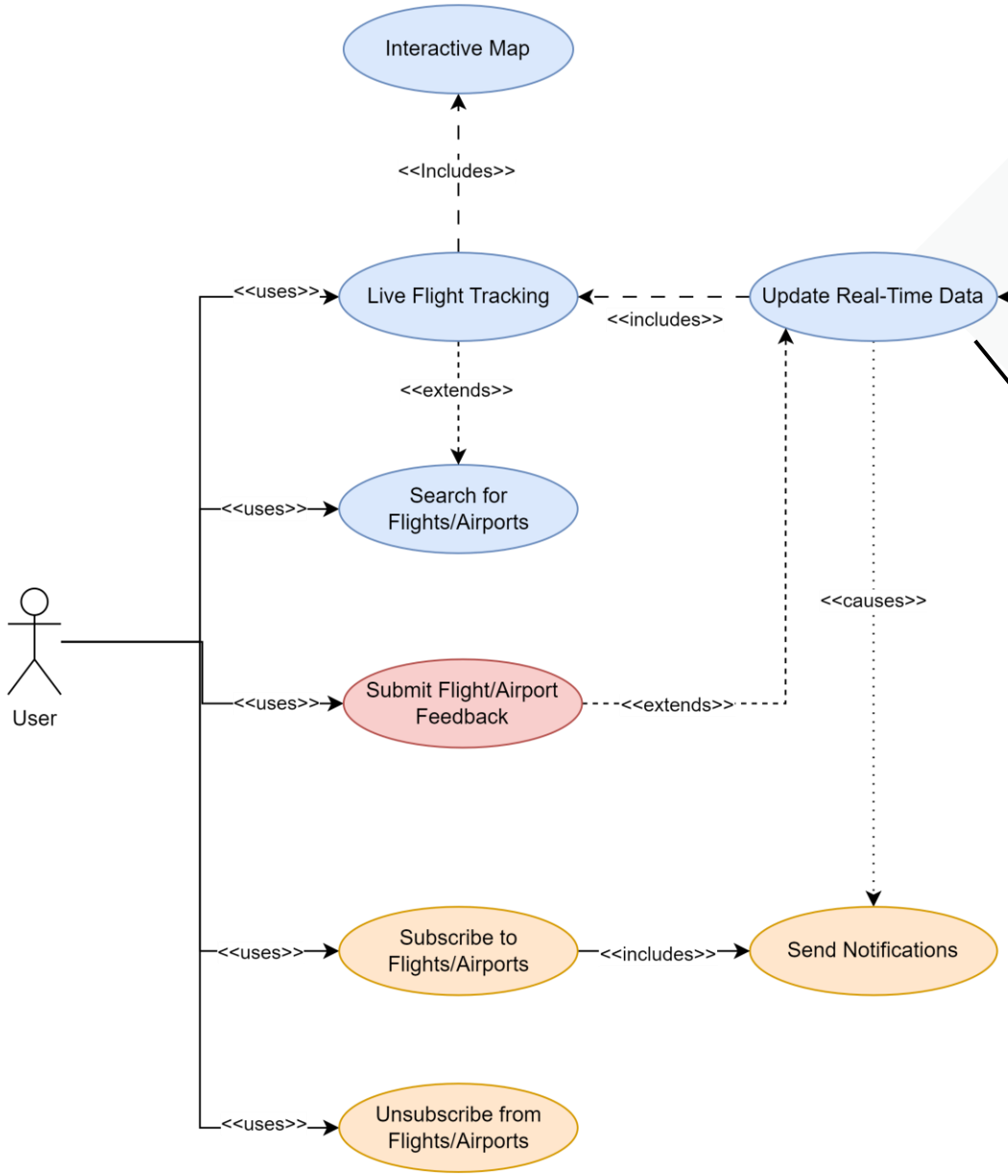
- **Adaptability**
  - The system shall be hostable on different platforms.
  - The system shall be Containerizable.
- **Performance & Scalability**
  - The system shall support high throughput for external API Data Intake.
  - The system shall be scalable.
- **Availability & Reliability**
  - The system must be accessible with different browsers and operating systems.
  - The system shall have minimal downtime.
  - The system shall have no complete downtime
    - Only downtime of specific services should occur.
- **Compliance**
  - The system must comply with GDPR’s user data regulations.

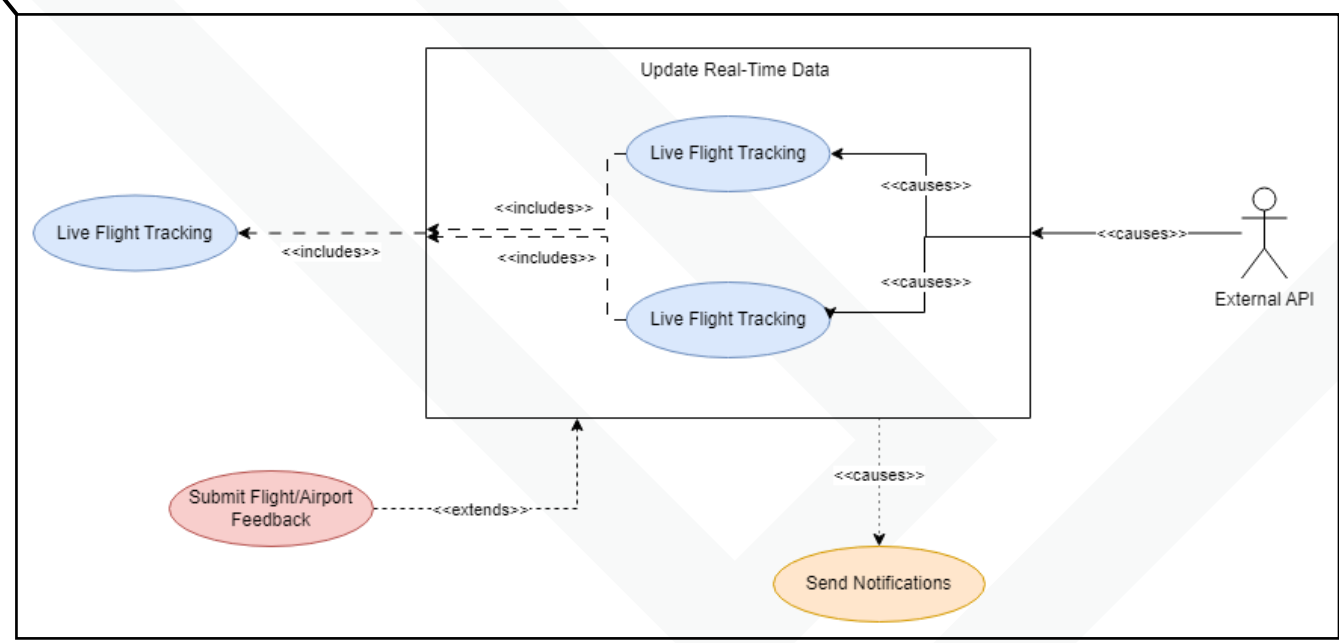
UML Us

Lancaster University Leipzig

co

diagr



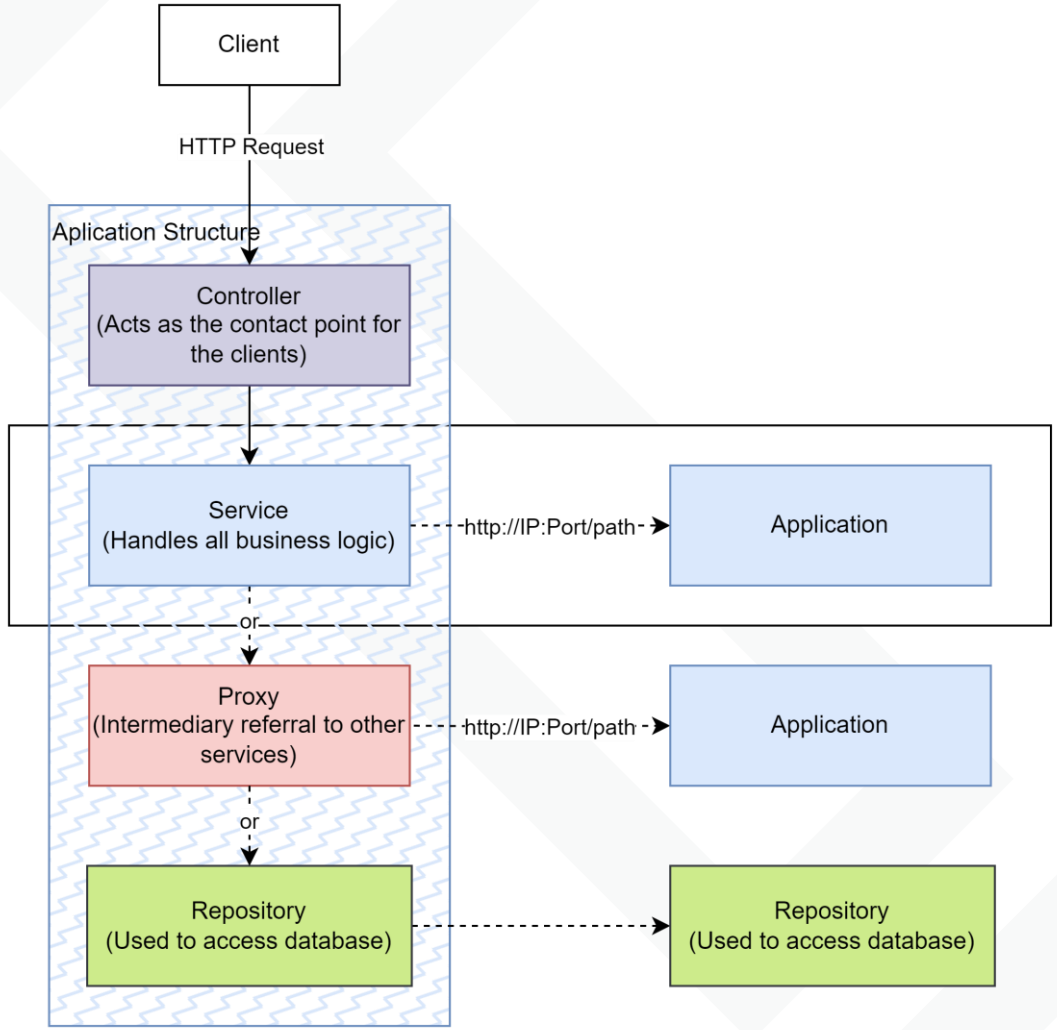


Lancaster University Leipzig

Software Design

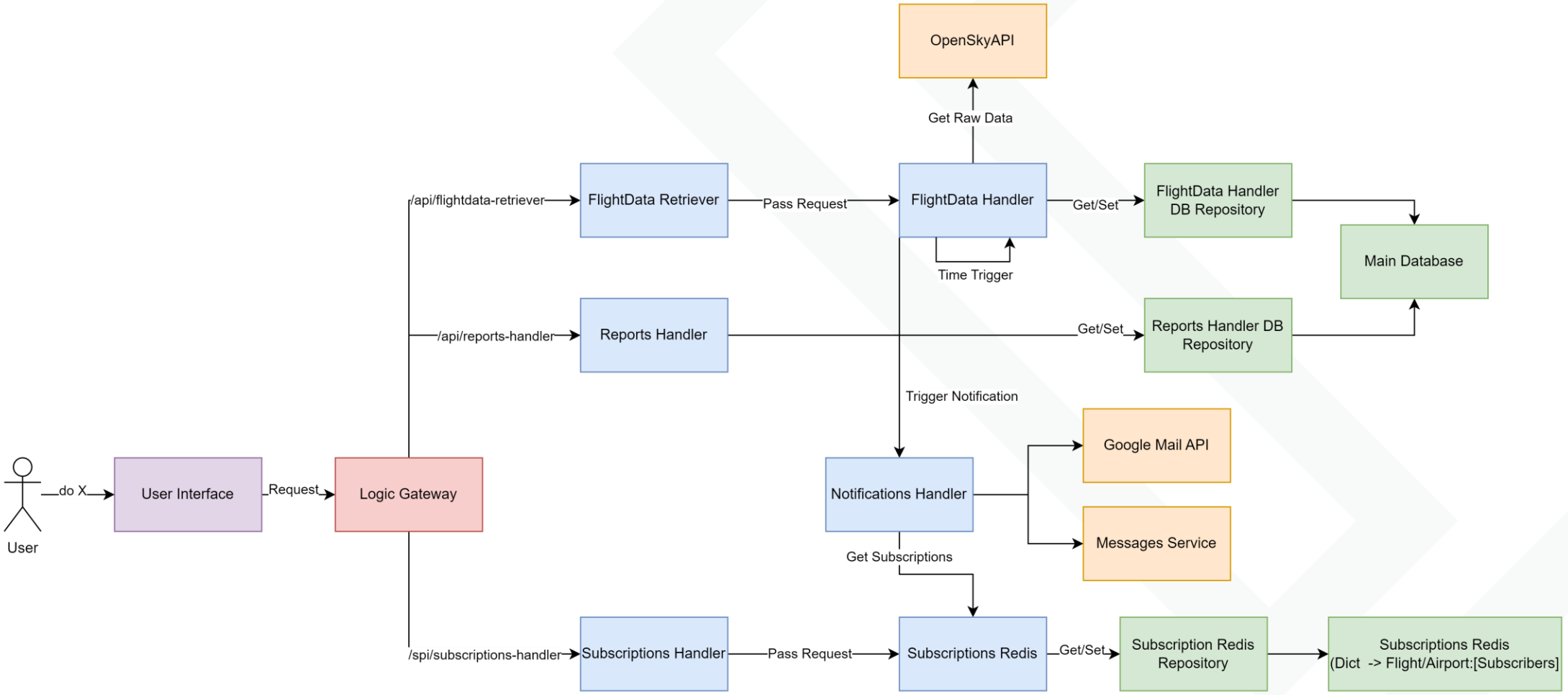
Architectural Style

- Layered Architecture
- Client-Server Architecture
- Proxy Pattern
- Repository Pattern
- Service-Oriented Architecture
- Database-Centric Architecture
- Microservices Architecture
- REST

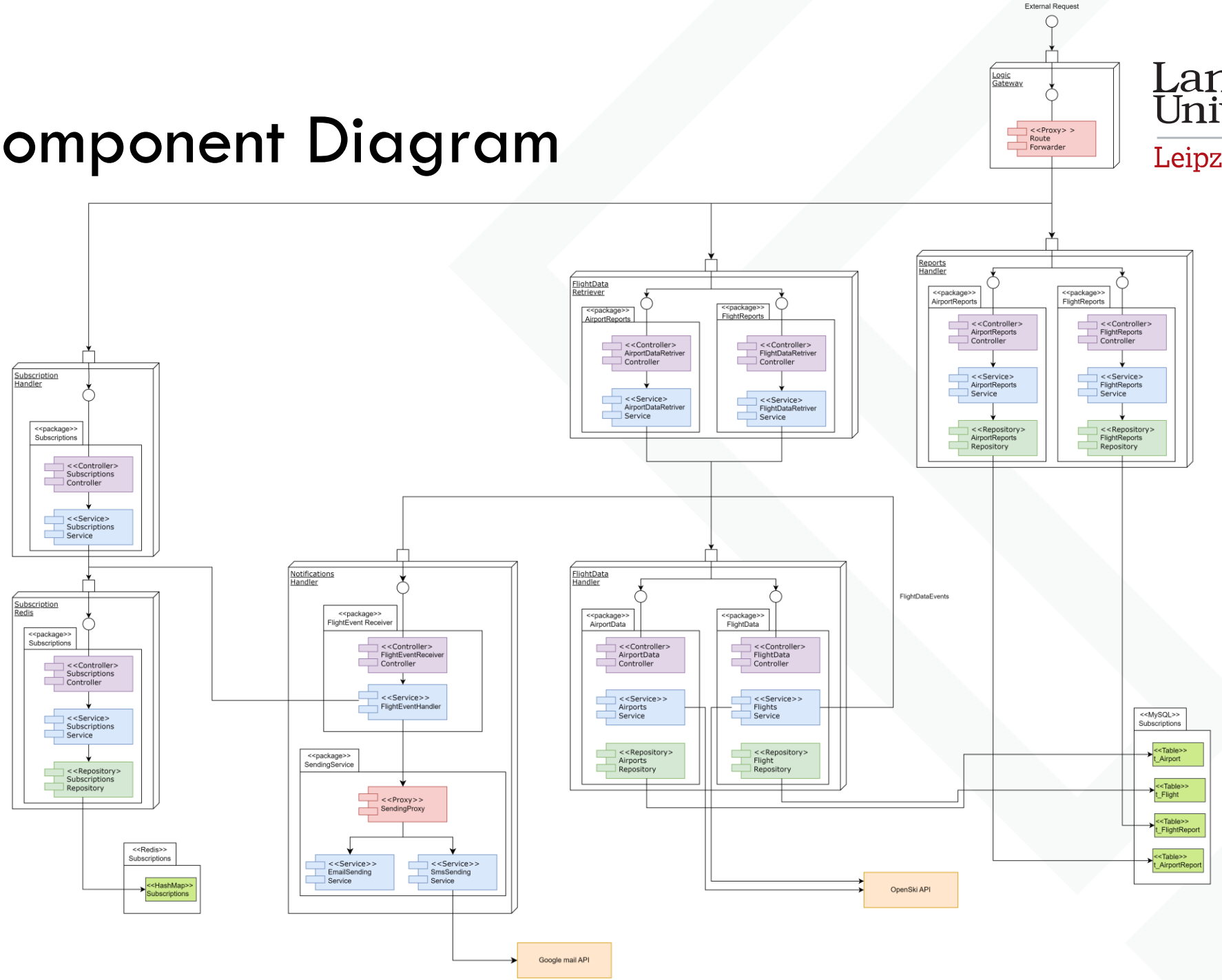


System Design

Lancaster University Leipzig

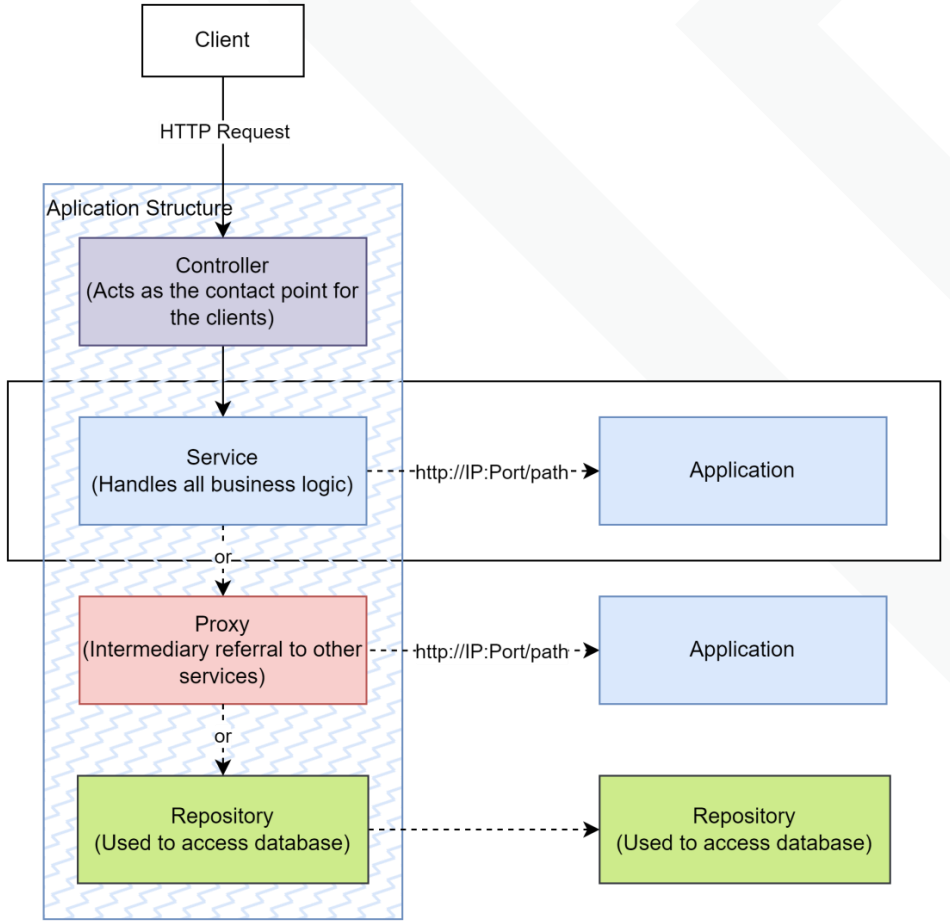


UML Component Diagram

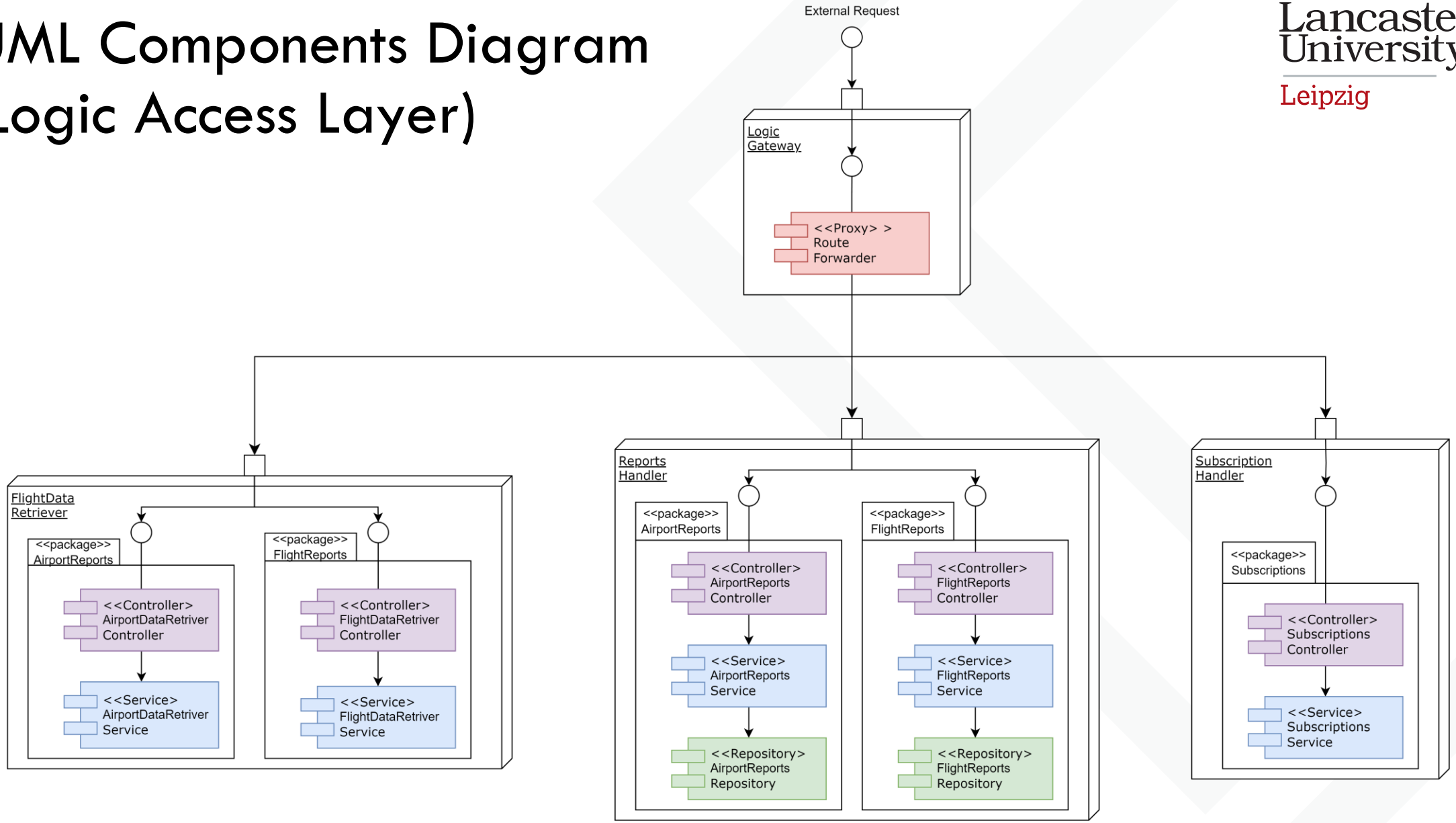


UML Components Diagram (General Service Design)

Lancaster University Leipzig

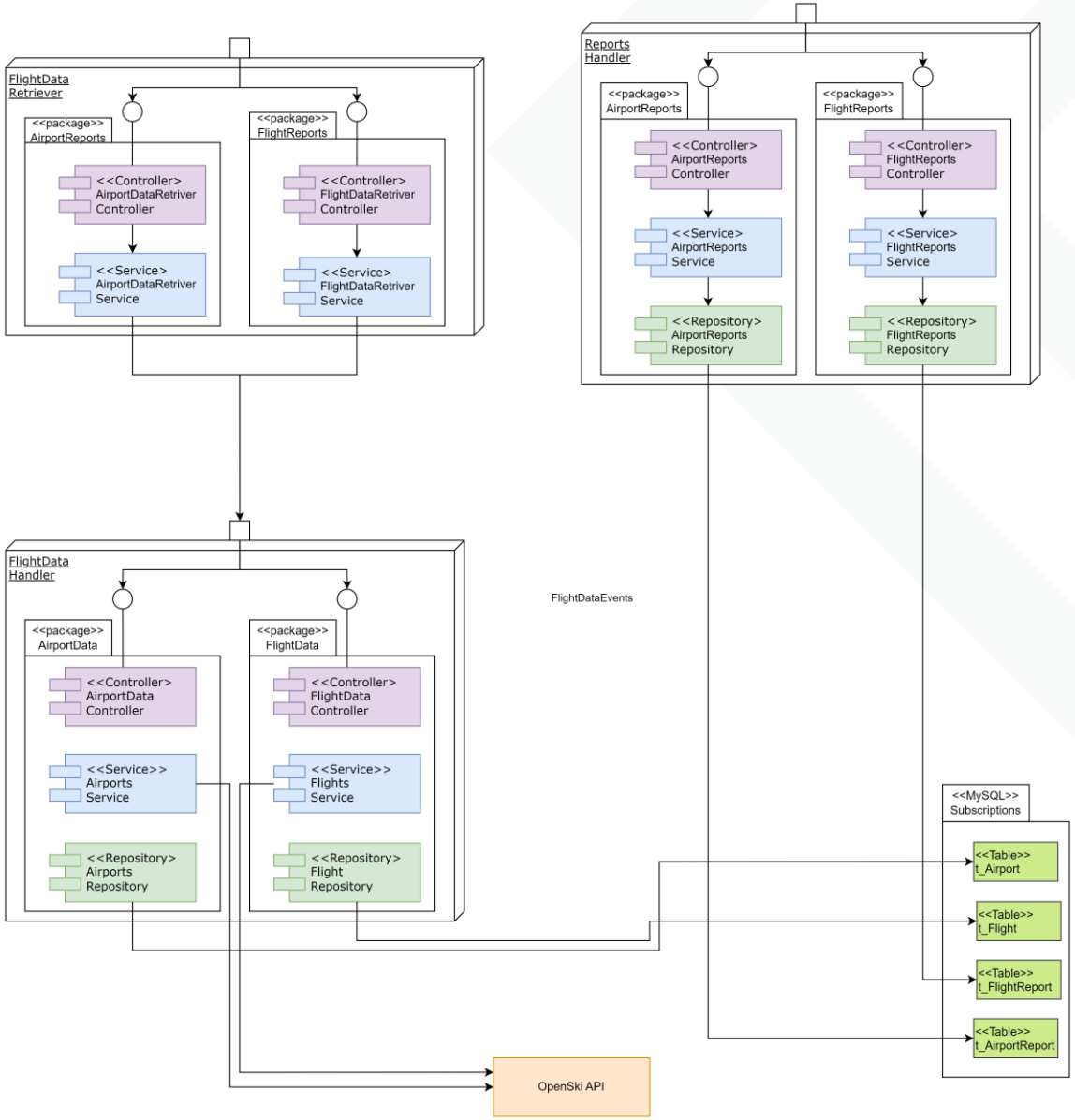


UML Components Diagram (Logic Access Layer)



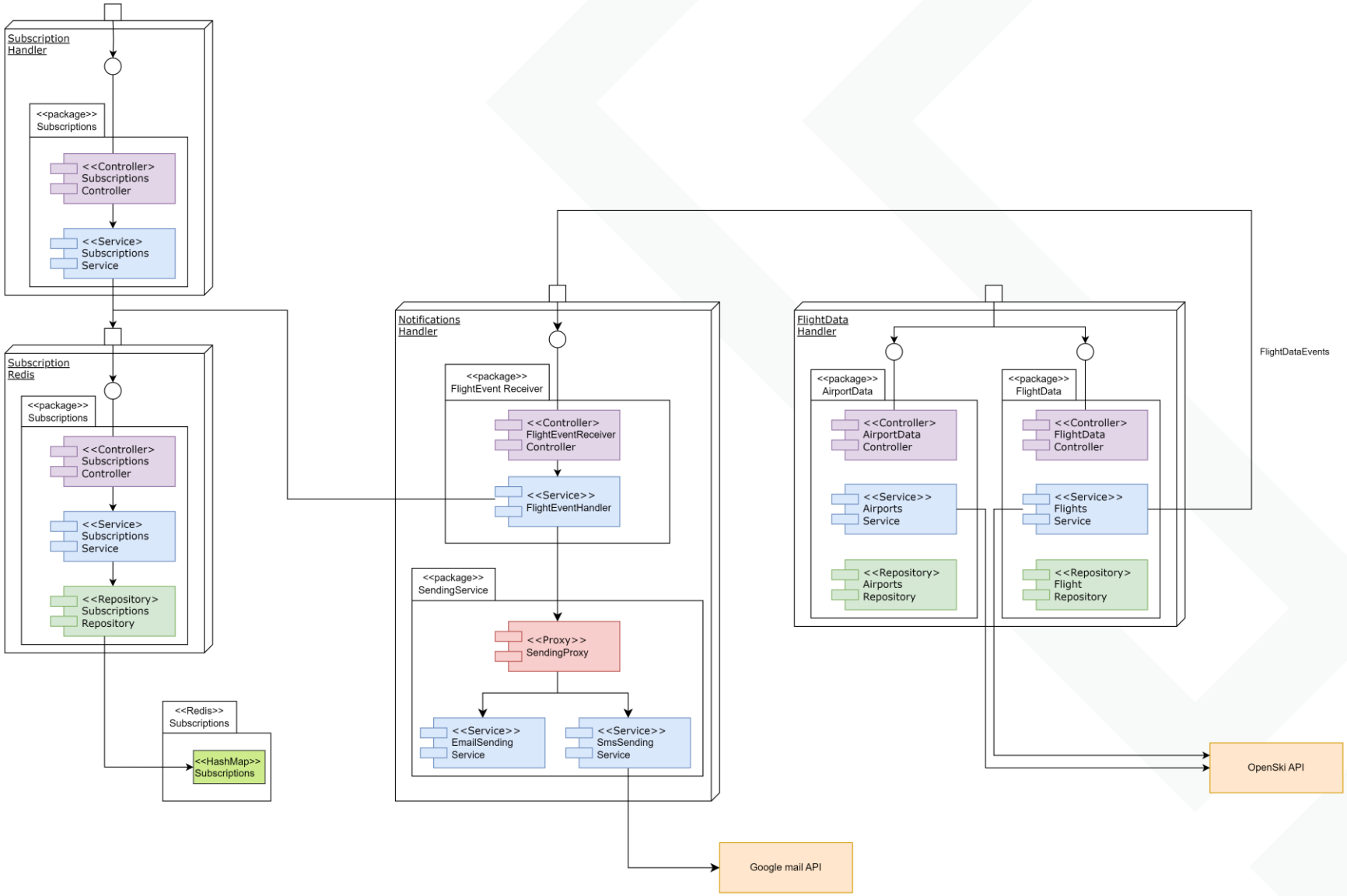
UML Components Diagram (FlightData System)

Lancaster University Leipzig



UML Components Diagram (Subscriptions System)

Lancaster University Leipzig



Lancaster University Leipzig

Software Implementation

Tools and Technologies

Lancaster University Leipzig

- Front-End:
  - Mapbox API
  - React
  - React Router-Dom
- Back-End:
  - MySQL
  - Redis
  - OpenSky API
  - SpringBoot
  - Gradle
  - Docker
  - Google Mail API

REST API Sample Calls

Lancaster University Leipzig

Submit new Subscription

```
HTTP
1  POST /api/subscription-handler/
       subscriptions HTTP/1.1
2  Host: localhost:10002
3  Content-Type: application/json
4  Content-Length: 89
5
6  {
7      "aviationDataID": "Test",
8      "name": "Random Name",
9      "email": "RandomEmail@gmail.com"
10 }
```

Status: 201 Created Time: 23 ms Size: 128 B

Status: 500 Internal Server Error Time: 17 ms Size: 293 B

Determine Subscription

```
HTTP
1  POST /api/subscription-handler/
       subscriptions/determine-subscription
       HTTP/1.1
2  Host: localhost:10002
3  Content-Type: application/json
4  Content-Length: 73
5
6  {
7      "email": "RandomEmail@gmail.com",
8      "aviationDataID": "Test"
9  }
```

```
1  true
```

```
1  false
```

Status: 200 OK Time: 931 ms Size: 168 B

Unsubscribe

```
HTTP
1  DELETE /api/subscription-handler/
       subscriptions/unsubscribe HTTP/1.1
2  Host: localhost:10002
3  Content-Type: application/json
4  Content-Length: 73
5
6  {
7      "email": "RandomEmail@gmail.com",
8      "aviationDataID": "Test"
9  }
```

Status: 200 OK Time: 22 ms Size: 123 B

Status: 500 Internal Server Error Time: 64 ms Size: 305 B

REST API Sample Calls

Lancaster University Leipzig

Get Airport /{IATA}

GET http://localhost:10020/api/airportDataController/getAirportByCode/BER


```
{
    "arrivals": [],
    "departures": [],
    "iata": "BER",
    "icao": "EDDB",
    "airport_name": "Berlin Brandenburg Airport",
    "region_name": null,
    "country": "DE",
    "latitude": "52.3514",
    "longitude": "13.4939"
}
```


Status: 200 OK Time: 34 ms Size: 2.38 KB

Get Flight By Callsign /{callsign}

GET http://localhost:10020/api/flightController/getFlightByCallsign/DAL624


```
{
    "flightCode": "DAL624",
    "airline": "Delta Air Lines (USA) Delta",
    "flightDepAirportCode": "MEX",
    "flightArrAirportCode": "JFK",
    "flightDepTime": "2024-03-13T08:50:06.000+00:00",
    "flightArrTime": "2024-03-13T02:00:00.000+00:00",
    "flightDepAirport": "Mexico City International Airport",
    "flightArrAirport": "John F. Kennedy International Airport",
    "flightDepExpDelay": 0,
    "flightDepLatitude": "19.4363",
    "flightDepLongitude": "-99.0721",
    "flightArrExpDelay": 0,
    "flightArrLatitude": "40.6397",
    "flightArrLongitude": "-73.7789",
    "flightLatitude": 30.841,
    "flightLongitude": -90.5455
}
```


Status: 200 OK Time: 734 ms Size: 714 B

Get All States (Flights)

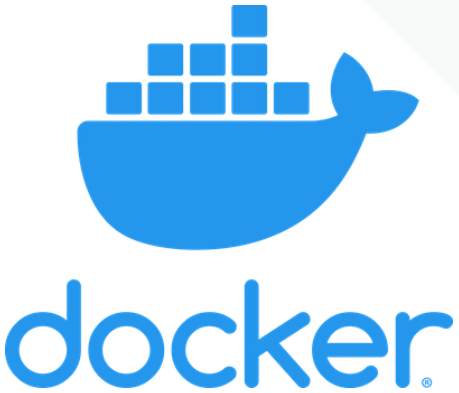
GET http://localhost:10020/api/flightController/getAllStates

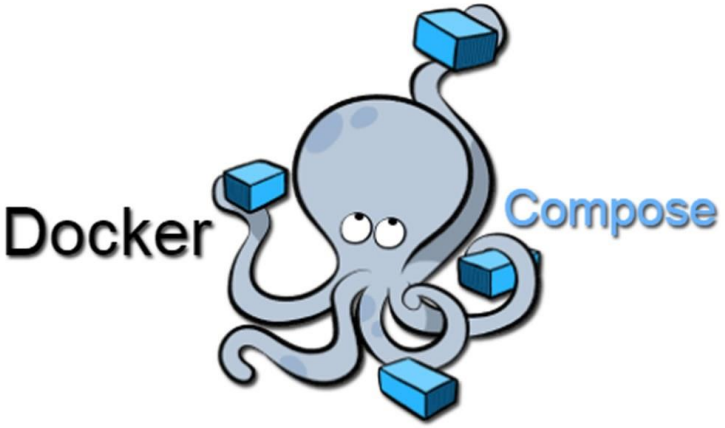

```
{
    "icao24": "ac96b8",
    "callsign": "AAL1346 ",
    "airline": "American Airlines (USA) American",
    "origin_country": "United States",
    "time_position": "1970-01-20T18:57:19.000+00:00",
    "last_contact": "1970-01-20T18:57:19.000+00:00",
    "longitude": -71.1024,
    "latitude": 21.572,
    "baro_altitude": 11277.6,
    "on_ground": false,
    "velocity": 260.77,
    "true_track": 166.07,
    "vertical_rate": 0.0,
    "sensors": null,
    "geo_altitude": 11757.7,
    "squawk": null,
    "spi": false,
    "position_source": 0,
    "category": 0,
    "status": "Flying",
    "lastTimeUpdated": "2024-03-07T20:22:57.777+00:00",
    "isDepartureDelayed": false,
    "isArrivalDelayed": false,
```


Status: 200 OK Time: 2.58 s Size: 14.96 MB

Deploying Application

Lancaster University Leipzig





Lancaster University Leipzig

Software Testing

Unit Testing - doSearch()

Lancaster University Leipzig

1. Running & Reading Flight from Python
   1. Between ~9K to ~20K objects
2. Turn it into JSON
3. JSON airline assignation, validation and sanitation
   1. Issues found throughout testing and their fixes made their way to the final product.
4. Database upload

Expected API Output:

```
{
    'baro_altitude': 7848.6,
    'callsign': 'ACA1074 ',
    'category': 0,
    'geo_altitude': 7688.58,
    'icao24': 'c067ae',
    'last_contact': 1710277134,
    'latitude': 46.1471,
    'longitude': -75.5616,
    'on_ground': false,
    'origin_country': 'Canada',
    'position_source': 0,
    'sensors': null,
    'spi': false,
    'squawk': '5236',
    'time_position': 1710277134,
    'true_track': 87.96,
    'velocity': 231.13,
    'vertical_rate': -9.1
}
```

ACA1074

Airline DB entry:

IATA	ICAO	Airline
AC	ACA	Air Canada (Canada) Air Canada

Final flight DB Object:

```
{
    'icao24': 'c067ae',
    'callsign': 'ACA1074',
    'origin_country': 'Canada',
    'time_position': '2024-03-12 21:32:24',
    'last_contact': '2024-03-12 21:32:24',
    'longitude': -75.5616,
    'latitude': 46.1471,
    'baro_altitude': 7848.6,
    'on_ground': false,
    'velocity': 231.13,
    'true_track': 87.96,
    'vertical_rate': -9.1,
    'sensors': null,
    'geo_altitude': 7688.58,
    'squawk': '5236',
    'spi': false,
    'position_source': 0,
    'category': 0,
    'airline': 'Air Canada (Canada) Air Canada',
    'status': 'Flying',
    'last_time_updated': '2024-03-07 20:22:57.777000'
}
```

To DB:

Unit Testing – search airport code

Lancaster University Leipzig

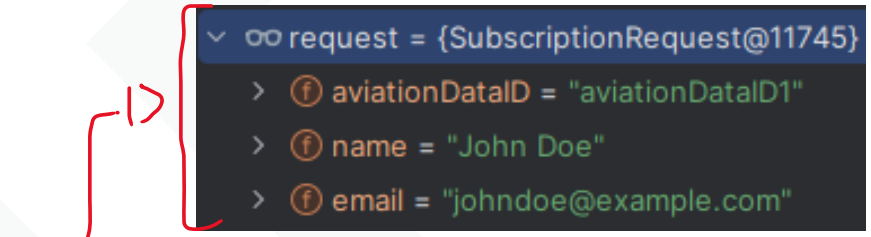
1. The test will introduce the value ‘MEX’ and it should redirect to the website of the information of the Airport.
2. Displays the information of ‘MEX’ at the Airport page.

```
    // Check if the mock was called with the correct arguments
    expect(mockedUseNavigate).toHaveBeenCalledWith( params: '/ATAirportPage', {
        replace: true,
        state: { IATA: 'MEX' },
    });
});
```

Integration Test – Subscription Lifecycle:

Lancaster University Leipzig

1. Start & Configure Redis TestContainer
2. Run SubscriptionLifecycle Test
   1. **Save** new SubscriptionRequest through Service.
   2. **Retrieve Subscriptions** through Service with SubscriptionRequest's AviationDataID.
   3. **Validate** that SubscriptionRequest was stored.
   4. **Unsubscribe** Email from AviationDataID through Service.
   5. **Validate** that Subscription is no longer stored.



To DB

From DB

Recap

Lancaster University Leipzig

Questions

Lancaster University Leipzig