Solution Architect Challenge

1. Introduction

# First, I would like to thank you all for giving me an opportunity to appear in an interview process in one of the leading airline carriers. As stated in shared problem challenge document, I tried keeping all information regarding challenges and problem statements.

There are basically 3 main sections in problem statement document

# **Requirement Breakdown:** Under this statement, I could see there are two problems mentioned and I have updated the same along with details in the same document under section 2.

# **Architecture Breakdown:** I could understand that there is monolithic application and booking service is huge legacy system, rest all should be breakdown to microservices. I assume based on my experience what could be major services and tried breaking the same in microservices which are listed and explained in the same document under section 3.

# **Programming Exercise:** Programming exercise have two parts of it, writing REST API for status service and collecting event and show them in elastic search and Kibana dashboard. I did his exercise, written sample program i.e., REST service and logging few businesses event, these events would be display in Kibana dashboard through Elastic search and same has been explained under section 4.

1. Requirement Breakdown

# Below screenshot has been attached in problem statement and questions mentioned in the same below.

# Graphical user interface, text, application, email Description automatically generated

* 1. A. What question would you asked to understand requirement better?

# **What I understand here:** Product owner wants digital team to work on above problem where end user should be able to search flight for given origin, destination, flight departure and return flight date.

# As an Architect when I deep dive into requirements there will be many questions in my mind and below are few samples

## Q1: What should be search trigger point?

# **Two-character search** - the moment user has typed two-characters, search should start, or search should be expecting minimum three character. For example, user start typing of FR – there could be many airports name and code (city name and city code as well), which may start with FR and if this is the requirement then we should start searching when text box identifies two character in the field.

# Example – when user type LO then there would be many options like Los Angeles (LAX) and London (LON) again London has 5 airports in it then search may give LAX first and then LON followed by all airports (LHR, LGW, LTN, STN, SEN) etc. and search will continue with the same and when user type 3 character then most specific information could be delivered.

# **Three-character search** - if search must look for 3 characters, then it will be easier for API/Services should return most appropriate result like as per IATA any airport and city code must be of 3 character like FRA/ZRH/DXB etc. So, if user start typing and length reaches 3 characters then service should return specific information to relevant information like

# Example – if user type LON, then along with LON city all airports must be displayed

# Note - One result is arrived we can segregate then in Local/International Airports or National/International Airports (as part of additional requirement)

## Q2: Above questions remains same for Destination Airport

# Why: Q1 and Q2 will help us to build service which will be backbone for airport search, we can give more flexibility and maintain system behavior consist of and boost up the performance as well.

## Q3: Is there any restrictions business wants to keep like if carrier flies to 20 airport/cities then origin and destination search must be limited to that

# Why: Usually City and Airport are part of MDM (master data) if carrier operates in certain destination, then we can keep out or limit our search for those destinations to build performance in the application

## Q4: Should user always select outgoing flight date, or it should be defaulted to some date like today or earliest by tomorrow?

# Why: This question will help to avoid unnecessary calls, if users are restricted to select outgoing flight date, then search will be very specific to the users input

## Q5: What if user select past date in Outgoing flight date or Returned flight date (should be disabled)?

# Why: Usually answer will be yes as no carrier can list down the part flight, however, as part of the requirement it will be okay to ask as this will be UX experience

## Q6: Is there any option business wants to keep for One way flight (ONEWAY and ROUND trip)?

# Why: Since I don’t see in screen grab so I asked this question for better search results and service behavior may change based on business decision

## Q7: Return flight date will depends on Question6’s answer, if business says that we should have One way flight then Return flight date will be disabled else we should force user to give desired date

# Why: Retuned flight date will be depending on if we have ONEWAY or ROUND flight option and what is selected by default, this question will help if user should force to feed this details or system will take decision to auto disabled

## Q8: What should be default search options like earliest arrival OR earliest departure (fastest route or shortest route)

# Why: There are many search options like earliest departure, earliest arrival etc. this input will help at service end to take decision and display the result.

## Q9: Travelers’s information, by default one adult is selected and that would be our default configuration for Trave class?

# Why: Travelers is select one adult be default and if we have any class defined then we need to default that as well like Economy/Business/First class etc.

# Q10: Is there any via airport user business wants to keep like user wanted to go to Dubai to London via Frankfurt like Dubai to FRA and FRA to LON some like this.

# Why: Asked this question to know if search API should have this feature build (it also known as Multi city option), user can depart from Dubai on 1st July and reach Frankfurt on 1st July and then can depart again from Frankfurt on 2nd July to LHR airport.

* 1. How would you define an appropriate Minimum Viable Product (MVP)?

# I understood that this question is limited to only flight search however, I will be mentioned in all journeys like flight search/booking/modify booking and flight status. As per my experience, I believe user should be able to feed below information and to keep best user experience, let system should take few defaults decision and once all details are populated then system should be able to populate result and end user should be able to select flight and proceed with the booking. Application should generate booking ID and PNR number and end user must allow to book as guest or login user in both persona

# Flight Search:

# Should be able to search departure airport and arrival airport with listed suggestion (up to 2- or 3-character type onward application should search)

# End user must be able to select both airports (Departure and Arrival)

# End user should be able to change outgoing flight date (today late or tomorrow onwards) and return flight as well.

# End user should be able to click on ‘Search for flight’

# Application must be populated with appropriate list of flight details

# Booking:

# End user must be able to select one flight from the listed options

# End user must be able to proceed booking with Guest User or logged in user

# End user should be able to select payment option (credit/debit card/ net backing or any other payment mode like UPI or apple pay)

# Application should be to booking flight for the user

# Once payment is cleared booking must be confirmed

# Modify Booking:

# End user must allow to able to search flight with PNR number or booking Id

# End user must be able to search flight based on preference

# End user must allow to proceed with the booking

# Booking Status:

# End user must be able to give flight number and Travel date to get the flight status

1. Architectural Breakdown  
    Text

   Description automatically generated

# The best way to define monolithic architecture is the application that grew over time became unmanageable and difficult to understand and had tight coupling and very complex to maintain architecture and application both. Microservice architecture solve the shortcomings of monolithic application, so most of the organization are moving towards microservices architecture

# Microservices architecture promote applications that consist of one or many independent services, where each microservice contains a business capability and have its contracts and own database/schemas. Each microservice has independent lifecycle, means can be developed in any language, deployed, scaled (up/down) and managed independently.

# There are couple of challenges we need to consider while working on microservice architecture

## Identify Services (Service Decomposition)

# Breaking monolithic application is not so easy, we need first analyze services properly in monolithic and identify component in monolithic which are loosely coupled and took them first candidate of micro service architecture

# There is component which will be more focused by the business and business wants to add more features (component enhancements) for such cases we need to target them as all future enhancements can be done once micro services are build

# We should avoid component enhancements or feature addition in monolith application

## Identify first candidate to build

# The component which is loosely coupled in monolith should be first candidate We should add more test coverage so that service can perform well in production

# Component should be high in demand so that any future enhancements can be done easily, and business can see beauty of microservices (scalability deployment cycle etc.)

# We should separate database part as well along with microservices

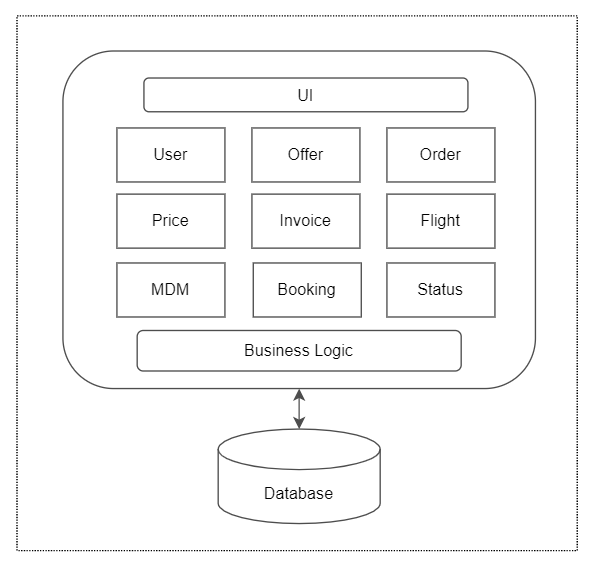
## Planning of Migration

# Once services are built, we will have monolith and microservices both should be deployed in production. We are plan for glue code which will connect monolith application to microservices to get the required information and once information is shared by the microservices then monolithic application flow will proceed further.

# Once this migration is working fine, we will have one migration strategy and other microservices can be planned in the same way

# 

* 1. A. Please visualize your understanding of the current system.  
       
     Monolith Architecture: As explained above, monolith will have all feature build in single application and same will be deployed in production and all features will be shared the same database as illustrated in below figure.



# As illustrated in above diagram, all services are built in one application and any feature and enhancement has been made within the same application and now it’s very difficult to manage the same. User services (passengers), Offers, Booking, Price, Invoice and Flight services are built in same application. Since this is airline application and as per IATA there might be changes and to adopt industry standards there would be many enhancements and changes which would have made in the same application.

# Problem with the current system

# It is very difficult to maintain

# Code base is increasing day by day

# Once application is down then all the services are down

# Deployment will take more time as monolithic

# Enhancement is difficult as each service are tightly coupled, any defect might impact other services as well

# Downtime may be more as if one deployment had issues then all services will be brought down

# Technology adoption is the challenge

# Quality check will take more time as all services are tightly coupled and QA team need to test complete application or bare minimum all the impacted area.

* 1. B. Please visualize your idea of a microservice architecture, that would split up our monolithic application. Please document your assumptions

# As explained in above section, we need to start identifying component which can be converted into microservices and slowly all components will be translated, and architecture will look something like below

# Diagram Description automatically generated

# As illustrated in above diagram, high level services identified in monolithic application can be identified as M/C as User, Booking, Status, MDM, Flight Offer, Price, Order, and Invoice (not mentioned in the diagram). Each microservices will have its own database and it will connect to the same, there are chances that one microservice can communicate with another microservice to fulfill the business requirements. Also, there are changes that one microservice wants to update data in another microservice, but any update will happen to microservices

## Assumptions

# Booking services, I have depicted in separate section

# Authentication and authorization will be happening though another microservices let’s call it as IAM (Identity and Access management)

# Usually, each microservices will have its own database if we have resource constraint, at least schema must be different

## Microservice Communication

# Microservice can communicate with each other as illustrated in below diagram Diagram Description automatically generated

* 1. Please visualize how the front end (browser, mobile App, voice assistant) consume this service

# Graphical user interface, application Description automatically generated

# Above architecture diagram is high level flow of API/services from the App, as illustrated any service call will be first generated from the app and app could be accessed via a browser (web browser, Mobile app) and since API is access over https then complete content will be encrypted and send. At premises, https will be loaded as soon it reached to the load balancer. App is going to call an API with load balancer (could be F5 or any other). This LB will have configuration for API gateway and Juul can be implementation of API gateway. Jull will have all configurations and service URL which is deployed in Trusted zone. Direct service URL will not be exposed to the client, means App cannot call direct service URL instead it will be through load balancer and then will be delegated via gateway.

# As mentioned in above diagram, once requested reaches to Jull, it will identify SD and through SB request will be delegated to the product services and these services are deployed in two difference VMs

* 1. D. How do you plan the roll-out of the new architecture on production?

# Once services are built, we will have monolith and microservices both should be deployed in production.

# We are plan for glue code which will connect monolith application to microservices to get the required information and once information is shared by the microservices then monolithic application flow will proceed further. Once this migration is working fine, we will have one migration strategy and other microservices can be planned in the same way

# **Existing Monolithic**: All services are running in monolithic applications; I have taken couple of services examples as below.

Diagram

Description automatically generated

# Design/business team has decided to move service4 to the microservice. Once Service4 is build, then we will have both Monolithic and Microservice4 running as below

# Diagram Description automatically generated

# Now, with IPC (Inter-Process Communication Adapter) we can call Service4 microservice from monolithic application. Once this migration is success, then slowly we can plan for other component/microservices move to production.

* 1. Which frameworks and technologies would you use to implement your architecture?

# Since this I need to build in microservice architecture I would surely prefer Spring Boot and Spring cloud family. To register service o registry would be preferred from Eureka library i.e., Service Discovery.

# Microservices configuration will be available and accessible through configuration server and spring cloud family have better implementation for the same. Rest all technology stacks are listed in below table.

# I did not mention specific version as it may change during implantation. Language would be Java1.8 or higher like java11 and Angular at front end.

|  |  |  |
| --- | --- | --- |
| S. No. | Function | Technology Component |
| 1 | Application Architecture | Microservices |
| 2 | Server Runtime | JDK (java Development Kit) |
| 3 | Front End | Angular |
| 4 | Service Registry | Eureka |
| 5 | Configuration Server | Spring Cloud Config |
| 6 | Database | Postgres |
| 7 | API Gateway | Zuul |
| 8 | Webserver | Tomcat |
| 9 | Runtime Environment | Docker Container |

# 

# 

Dealing with Booking Service – Legacy Booking Service

# As mentioned in the document booking part is the legacy system and cannot be replaced by microservices, to achieve this we can still write one microservice called ‘Order’ (any name we can suggest) and this microservice will call another service which is exposed in monolithic application. Order microservice will completely depends upon monolithic application availability. Service call and flow I have mentioned in below diagram

# Diagram Description automatically generated

# Graphical user interface Description automatically generated

# As explained in above section, I have analyzed how many high levels of microservices could be, now if we start building flight search service then service must get available flight along with price associated with it. Hence came across below logical architecture view

Flight Search M/S Logical View

# Diagram Description automatically generated

# As illustrated in above diagram, offer microservice will be responsible to fetch flight and price associated with it, stich the result and return the same to the UI component

# There are chances that Flight and Price both services are available in monolithic application then same can be exposed as REST API and this will be consumed by other flight and Price microservices

# Second option is, we have Flight and Price services are separated out from monolithic and these services are available as microservices and same can be directly used by the offer service

# First proposal is good if we must deliver product quickly, however, end goal must be second option

# I mentioned high level offer service input in diagram but if we sketch out JSON then, there will be many properties required to keep in the input

Graphical user interface, application

Description automatically generated

# Flight Status Service should accept flight number and Travel date and return flight information as mentioned in the problem statement.

# I believe this status will be maintain by either airline or any ground handler, in any case status must be updated in the flight resource.

# Status Microservice should get flight information from Flight microservice and if flight is handled by ground handler, then Status service may connect to Operation service as well to get the latest status. Once Status services have all required information, then it can process the data and return the result

Flight Status M/S Logical View

# Diagram Description automatically generated

# As illustrated in above diagram, status microservice should get flight information from flight service like expected/actual flight departure date, expected/actual flight arrival date, flight route and status like on time or delayed

1. Programming Exercise

# Text Description automatically generated

# As mentioned in exercise A, I have written one Status service which accept flight number and departure date as an input. If any flight available with request input, API will return success response and success response could be flight not found or proper flight details. I have used Sprint Boot to write service and documented it with swagger implementation.

# Git Repository: <https://github.com/manishkushwaha1412/eurodigital>

# Application is available in below Git repository and attaching Readme.txt to follow steps to run the same.

# Once application is build, docker image is created and executed, application will be app and running

# **How to build**: go to project directory and hit mvn clean install (Assuming, Maven 3.6.\* version is available with java 1.8)

# Graphical user interface, table Description automatically generated

# **Create Docker image and verify:** Text Description automatically generated

# Above step is already given in Readme.txt file.

# **Run docker image:** Graphical user interface, text Description automatically generated

# Once application is up and running, hot URL <http://localhost:9090/swagger-ui.html#/>

# Above link will open swagger documentation, which has list of APIs build.

# Graphical user interface, application Description automatically generated

# Note: I have written DDL and DML as well, which push some dummy data while server startup itself. API will be using the same data and display in result.

# Sample Outputs

# Success:

# Graphical user interface Description automatically generated

# Failures:

# Graphical user interface, text, application Description automatically generated

# Graphical user interface, text, application Description automatically generated

# Graphical user interface, text, application Description automatically generated

# 

# As mentioned in exercise B, I am using Spring Boot service and through light status I am triggering few alerts like which API is being called and what is the response of the API along with API input information.

# This information is getting pushed from application to Elastic search through Logstash and then showing the same events in Kibana. As mentioned, I have used elastic search and Kibana through docker images and used docker-compose to build and run images. Below are few screen grabs for more information.

# Graphical user interface, text Description automatically generated

# A screenshot of a computer Description automatically generated

# Graphical user interface, text, email Description automatically generated

# Graphical user interface, text, application, email Description automatically generated

1. Assumption

# I have explained all required information and tried documenting it as per my best knowledge and information. I understand there will be many gaps, and this could be filled once I get the proper requirement. I have mentioned tech stack as per my architecture views. In programming exercise section, I have mentioned what tech stack I used, I believe if someone running this program, they must have maven and java mentioned version installed in machine.