2.2 Airbnb / Booking.com system design

What is an Airbnb and how does it work?

What Is Airbnb? Airbnb, as in “Air Bed and Breakfast,” is **a service that lets property owners rent out their spaces to travelers looking for a place to stay**. Travelers can rent a space for multiple people to share, a shared space with private rooms, or the entire property for themselves.

How is Airbnb different from hotels?

One of the biggest differences between an Airbnb and a hotel is the **dining experience**. Airbnbs come with kitchenettes or kitchen, so it's easier to prepare your own meals. You can save big bucks on dining outside and create beautiful memories preparing your own food and sharing them with your loved ones

What is the purpose of Airbnb?

Airbnb is an online marketplace that **connects people who want to rent out their homes with people who are looking for accommodations in specific locales**. Airbnb offers people an easy, relatively stress-free way to earn some income from their property

Graphical user interface, text, application

Description automatically generated

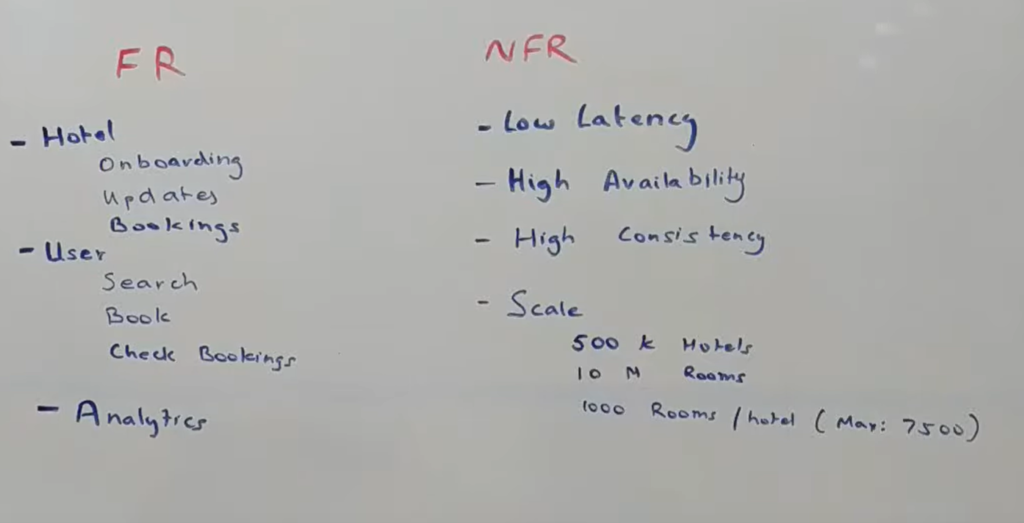
**objective**

Let see how to design a hotel booking system something very similar to **booking.com or airbnb**

**We have** two major consumer of this application

One is hotel side user and there are consumer who want to book the hotel

So for the hotel manager we have these three major functionalities



**From manager point of view**

1-: they should be able to onboard onto our platform

2- they should be able to update their property for example they might want to add room, they

they might want to change pricing, might want to add new images like that

3) they should be able to see that what all bookings are there

**From  user point of view**

**1-** they should be able to search a property in particular location with a couple of search criteria example :- they might want to filter within a price range or some aspect of property like five star property

2:- they should be able to book the hotel. Once they booked they should be able to look at their booking

**NFR**

**What is Low Latency?** × Low latency describes **a computer network that is optimized to process a very high volume of data messages with minimal delay (latency**

**1-** it should be run at very low latency

2- high availability

3- high consistent : if a user is booking the hotel , he should be able to see the hotel immediately

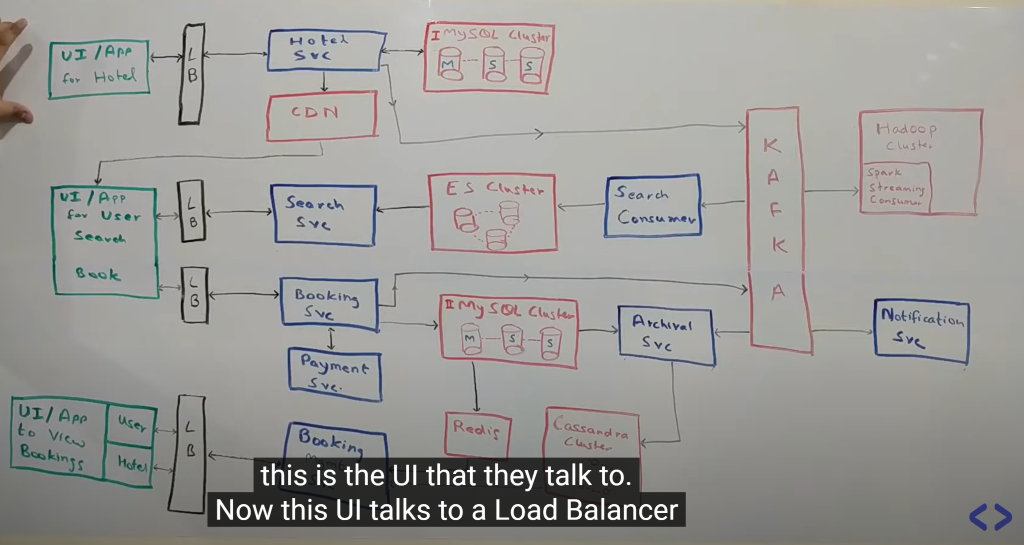
Scale point of view :

Google search tell that there are roughly  5 lakhs hotel in the world

There are roughly 12 millions room in all the hotel

Roughly assume that 1000 room in a hotel

**Let see whole flow**



Diagram

Description automatically generated

**Hotel flow**

**Basically** a UI will give to hotel manager through this it could be either a website or mobile app or any modification will be done by this UI

This UI will talk to load balancer. Through (LB) which it talk to hotel service.

**1-Hotel SVC :** this is basically a service  which manage the hotel part which is basically the onboarding  and the management.Now let say there is spike in traffic so there

Could be multiple nodes of this hotel services that could be added here and this becomes a horizontally scalable component

2-Now hotel data itself is very much relational data. Number of photos is not too many so it doesn't have even scale problem so we will be using clustered mysql here with one master and multiple slaves. Slaves can be added as when required.

Let say there huge spike in read traffic we can add more slaves but this data resides within mysql database.

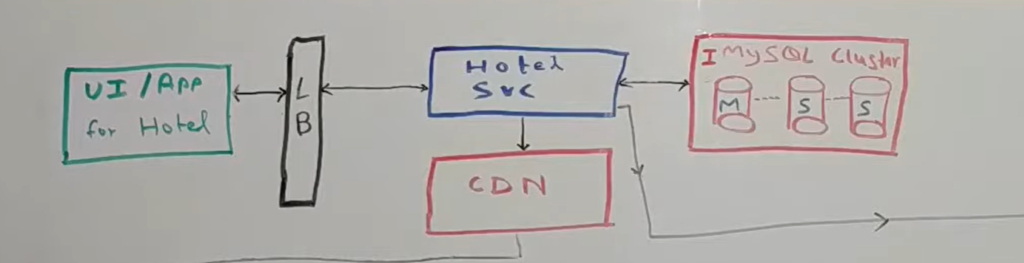
**3 CDN** - : now let say any images added . It can be of room , hotel building all those images would be stored into CDN and the reference to the CDN which is basically a URL of the image would be store in the d/b and that URL would be set to the customer

And whenever they want to render any image that can be directly look up from CDN

What is CDN?

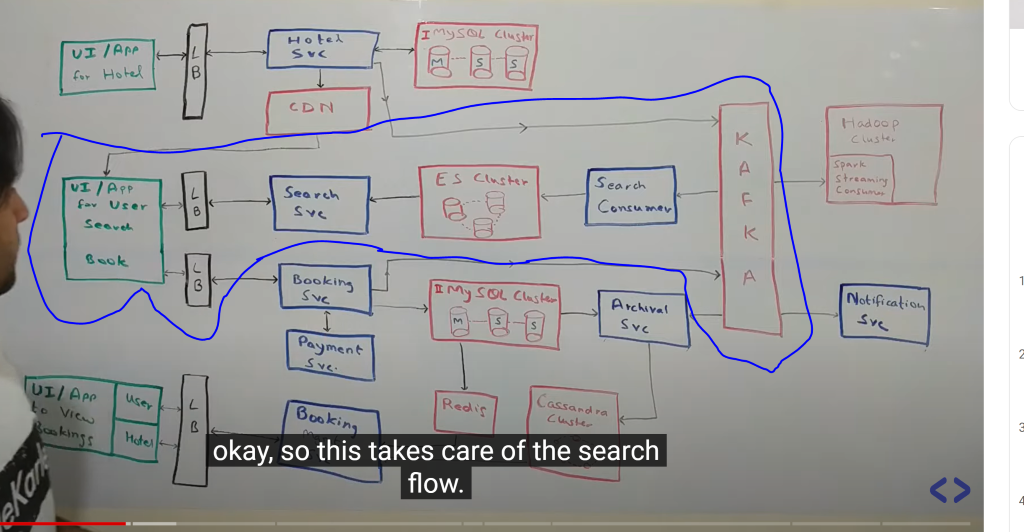
**It is basically geographically distributed data store which we will be using for sending out the images throughout the whole world**

Let say i m connecting from india and someone connect from usa they want to lookup image for a particular hotel so i will lookup on CDN server which is in india and other person will look up the cdn server which will be in us so this become hotel life cycle management



The next thing is basically let just say each time a modification is happening to the hotel. Let say new hotel come in and now we want to send this information to user  who want to search this hotel so there are multiple ways in which we can send out  this information to the search piece

**Search flow**



So each modification happening  within hotel service will flow through kafka cluster and there will be a multiple consumer will be sitting on top of this cluster **which will populate their data store for serving the search traffic. So** one of the consumer will be search consumer

What happens let hotel get a new room for example. There is a payload that is put into kafka which has all the information that is required. Now the search consumer will pull up payload from kafka and it stores it into it own d/b and this database would be use to power the search on the website. Now for search i m using elastic search.

Elastic search is basically a database that is built on lucene platform.

Similarly instead of elastic search we can also use solr here. Both are kind of similar component. The idea of using elastic search is to support fuzzy search

**A fuzzy search searches for text that matches a term closely instead of exactly. Fuzzy searches help you find relevant results even when the search terms are misspelled.**

**So all the** data of each individual hotel flows through the kafka via the search consumer into this elastic search cluster.

Now on top of this elastic search the **search svc.**

**Now** again there is increase in traffic, i can increase the number of nodes in kafka cluster. I can increase the number of search consumer and o can increase the number in elastic search cluster so till now whatever we talked it again a horizontally scalable

A "horizontally scalable" system is **one that can increase capacity by adding more computers to the system**

**And** again coming to search service this is service. This is the service which power the search on the website.

So user talk to   through load balancer to the search service whenever they want to search for a particular hotel and again they will give data range and location for example as a search criteria and along with that they can also provide some tags. Now those tags would be the properties of the hotel. A five star hotel is tag. A beach property is a tag. Now  search on elasticity would be happening on either  of these tags and the ranges that are provided basically the date range , price range etc.

===========

Diagram

Description automatically generated

**Book hotel**

Now once the user has seen some of the result on website they  would want to book a hotel. Now booking request again comes through load balancer to booking service.

Booking service again site on top of mysql database. Now these are two diff mysql cluster. Whenever booking happens it get stored into mysql database and booking service talk to payment service.

Normally what will happen a booking request will come.it store something it will send the request for payment once there is a success. It will mark the booking confirm.

Now again whenever booking is happening the data is flowing into the same kafka.

Why so let just say there was on room available in hotel right and that room is now booked i want to make sure this hotel is not available for search now in same date range because it not available so all of those information again sent to the same kafka

Which is read by search consumer and then it takes care of removing the hotel which is now completely booked.

Now if u can see there is something called archived service here. What i have done i m just storing live data into mysql. By live data i mean the booking that are done but not have been completed there by making sure that this is having a scale which is low enough that mysql can easily handle and once booking moved to terminal state so lets say booking is cancelled or booking is completed it will move through the archival

Service to a cassandra cluster. The reason i m using cassandra here is so cassandra is a very good d/b which can handle a huge amount of reads and writes. It has a constrains that it needs a partition key on which all the query should happen.

So let say if i want to search by booking\_id my partition key has to be booking\_id in that case i cannot do any kind of queries on a cassandra therefore i did not use cassandra as a source of truth database because on this d/b i need to do a large varities of queries.

But once it is archived we just need to do gets on those so cassandra make a good sense over here.now once the booking is done we need to notify all the people right.

So then comes the notification service. So lets say whenever booking is made or any changes are happening into a booking or it moves into a terminal state there will be a notification service that consumes events from this kafka and notifies the people so for example on each booking we need to notify the hotel. Whenever a booking is cancelled by the hotel we need to notify the consumer or on each booking we need to notify the consumer with an invoice so all of those is taken care by this notification service.

Now coming back to UI for hotels and users so each time a booking is done or even without that a user may want to see their old booking or a hotel might want to see all the bookings that they have. This is more only read only view for them. That will be powered by booking management system which talks now two data sources.

It talks mysql cluster for all the active booking which are to happen sometime in future

And to the Cassandra cluster for the booking that have already happened.

Now i m adding a Redis on top of this mysql to reduce the load on this mysql so redis will act as my cache and whenever i have a query so far  for example something get booking of a user so i can cache this result into this redis and it will be a **write through cache** so whenever a new booking is coming this will get updated. Now this is the functional flow.

The bigger component here is how do we do the**analytics on this.** So lets just say a businnes person want to know how **much revenue i m making or how many booking i m having or what are my best performing hotel and stuff** like that so they need to do a lot of analytics

Like while designing we never know what kind of analytics is required so what i have done for this i use a hadoop cluster on which i m pushing in all the events that are going into my kafka which is basically information all my hotels about all my booking, about all the transaction that happen in my system. So there will be a spark streaming consumer

That run somewhere that read from this kafka and put all the data into a hadoop cluster

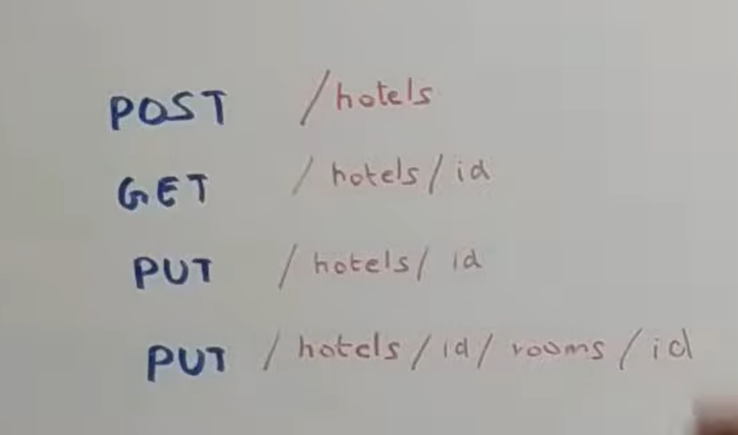
On which i can do hive queries or any different kind of queries and build a lot of reporting so this is overall how the system looks like and how the data flows and now get into the **details of the some of the component**

**==**

**Now** let look what hotel service internally is so it not a very complicated service it is basically a crud service which provides create , update, read , delete operation on the hotel data store.

There is lot of more thing but this will give the feeling of how it should be.

**So look at some of the api**

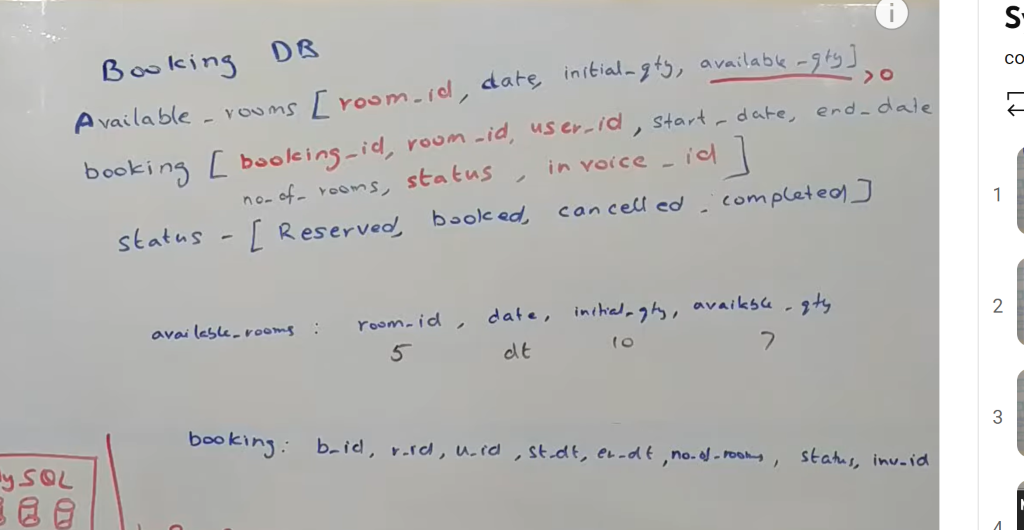


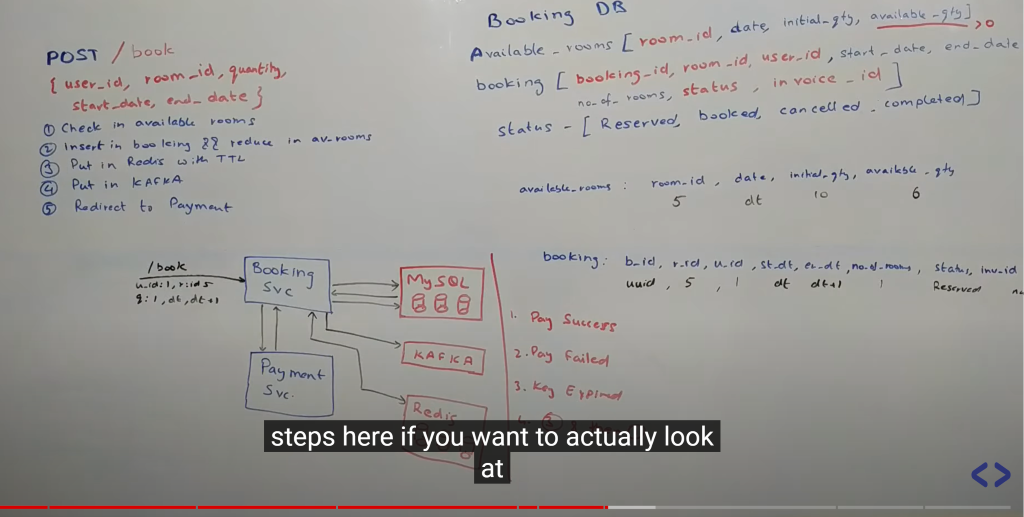
**DB schema**

Text, letter

Description automatically generated

**Now let looks at internal functioning of booking service**





**If room is** reserved by user but payment is not done so we need to make room available after 5 min so that someone else can booked it

There are multiple ways to do it one way is**time to live (TTL) of redis**

**W**ill put a key with timestamp into redis and redis have featured of callback. So whenever key is getting expired will get a notification.

**In case of payment** everything remain the same just status is **booked wll** get invoice

In case of payment failed booking status is cancelled and there would be no invoice

Ttl is not so perfect so we can use polling mechanism, in redis  but that will be additional cost so there will be continuously polling and will need to add more nodes into cluster.

**Optimization**:

Payment is success we know key will expire after some time in redis but we dont want to keep key in redis even if the payment failured we know that key will expire after 5 min might as well delete the key then and there so these are certain optimization

**Across the whole infrastructure we need to keep an eye how** my cpu usage percentage is, how my memory usage percentage is , how my disk usage for redis is, how my disk usage e for elastic search is all of these thing we need to monitor

Now monitoring can be done through graphana kind of tool on which i can set up a tool.

Suppose we put some threshold and the moment it cross the threashold team will get an alert and will came to know something wrong occur.

**Now how this application will be available in whole geography**

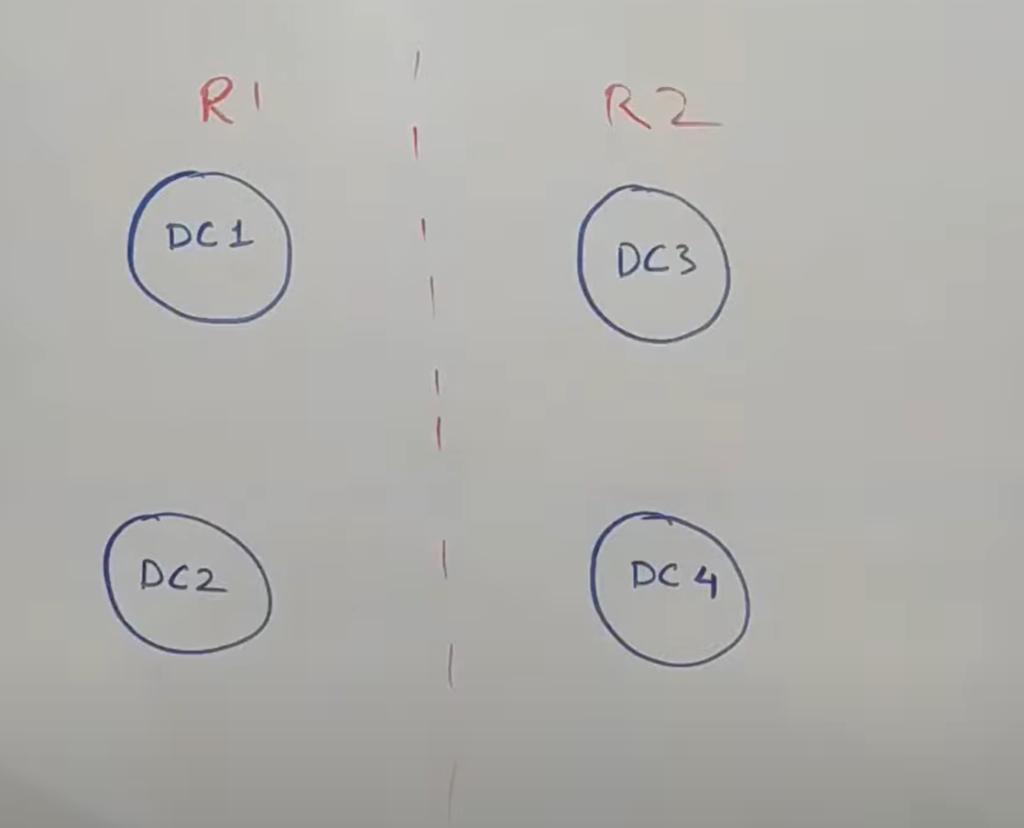
**Let** say there is a earthquake in one center how it will be available

Let say we have these 4 data center which are located in different geographical region across the globe

Dc1 is primary and all other are secondary DC and data is replicated in all the DC in real near time

But this is not good approach since all other three DC will be idle and we are using only 25 %

So what we can do we can divide the DC into region



So we can divide the data as per geography like india and usa**details of the some of the component**