



MERCEDES ASSIGNMENT

By

Mithlesh Singh Bisht

TABLE OF CONTENTS

Introduction.....	3
Problem Statement.....	5
Architectures used.....	6
Approach.....	7
Flowchart.....	8
Conclusion.....	9

Introduction

The **Mercedes-Benz S-Class** is a series of full-size luxury sedans, limousines and armoured sedans produced by the German automaker Mercedes-Benz, a division of German company Daimler AG. The S-Class is the designation for top-of-the-line Mercedes-Benz models and was officially introduced in 1972 with the W116, and has remained in use ever since. The S-Class is the flagship vehicle for Mercedes-Benz. The S-Class has debuted many of the company's latest innovations, including drive train technologies, interior features, and safety systems (such as the first seat-belt pretensioners).

The S-Class has ranked as the world's best-selling luxury sedan. In automotive terms, *Sonderklasse* refers to "a specially outfitted car." Although used colloquially for decades, following its official application in 1972, six generations of officially named *S-Klasse* sedans have been produced.



The Mercedes-Benz S-Class has 1 Diesel Engine and 3 Petrol Engine on offer. The Diesel engine is 2925 cc while the Petrol engine is 2996 cc and 3982 cc and 5980 cc . It is available with the Automatic transmission. Depending upon the variant and fuel type the S-Class has a mileage of . The S-Class is a 4 seater sedan and has length of 5462 mm, width of 2130 mm and a wheelbase of 3365 mm.



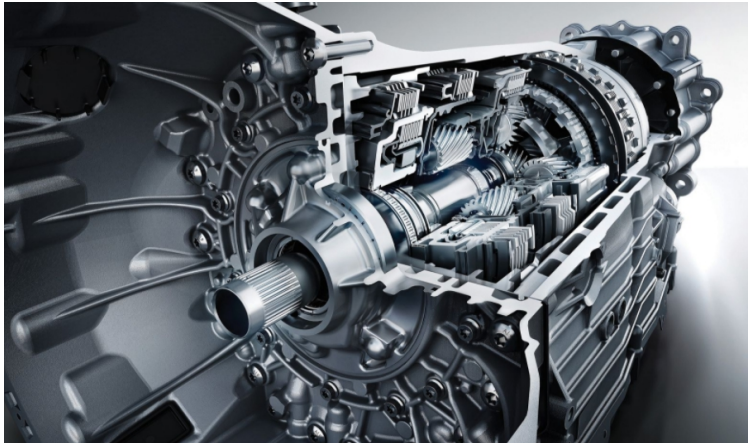
Mercedes-Benz launched the S-Class facelift on February 26, 2018. It is available in two variants: the S 350 d and the S 400, which are priced at Rs 1.33 crore and 1.37 crore (ex-showroom India) respectively.

The updated S-Class features quite a few revisions to its exteriors, mainly the headlamps, taillamps and bumpers. On the inside, the infotainment system gets an upgrade and now there are two 12.3-inch HD screens with a central command unit.

As far as rivals are concerned, the new S-Class goes up against the likes of the BMW 7 Series, Audi A8, Jaguar XJ and the Lexus LS 500h.

A few days back, Mercedes also launched the Maybach variants of the S-Class in India. The Mercedes-Maybach is offered in two variants: S 560 and S 650, which are priced at Rs 1.94 crore and 2.73 crore (ex-showroom India) respectively.

The S-Class packs the biggest change under its hood. It now gets a 3.0-litre inline-six, BSVI-compliant diesel engine that generates 286PS of power and 400Nm of torque. It is also offered with a 3.0-litre V6 petrol engine that makes 367PS of power and 500Nm of torque. Both the engines come mated to a nine-speed automatic transmission.



Powering the S 650 is a 6.0-litre AMG petrol engine that makes 630PS of power and 1,000Nm of torque, sent to the rear wheels by a 7-speed automatic transmission. The S 560, on the other hand, gets a 4.0-litre V8 petrol developing 469PS of power and 700Nm of torque. It is coupled to a 9-speed automatic transmission.



Mercedes-Benz S-Class is currently available in Diesel and Petrol engines. The 3982 cc Petrol engine generates a power of 603bhp@5500-6000rpm and a torque of 900Nm@2750-4500rpm. The 2925 cc Diesel engine generates a power of 281.61bhp@3400-4600rpm and a torque of 600Nm@1200-3200rpm. Mercedes-Benz S-Class is available in Automatic transmission only. Mercedes-Benz S-Class has a boot space of 500 litres and ground clearance of 109 mm. The kerb weight of S-Class is 2245 Kg. In configurations, Mercedes-Benz S-Class has a dimensions of 5462 mm in length, 2130 mm in width and 1498 mm³ mm in height.

Problem Statement

Design and implement an event driven service,

where an event of either true or false along with a random city of India is constantly pushed from one service to another service. Events are pushed once in every 2 minutes unless there is any update, if there is any update then the event is pushed instantly. The event receiving service should implement a messaging broker for handling the events and based upon the event value the service should do a specific task. Here the event true means that the fuel lid of the car is opened and false means its closed, when you get true then call an API to know the fuel cost at a given city (for instance {fuel lid: true; city: Bangalore}) and assuming that the car can take 1 litre/30 seconds then calculate the amount of fuel got into the tank and calculate the price. For now, price can be logged inside the event receiving service. Car location can be random, for any random city try to get the fuel price from a 3rd party provider and calculate the amount of fuel cost. Explore an 3rd party API which can provide fuel cost of any given Indian city, if you were not able to find out the API then create a mocked service that returns fuel cost for a given Indian city. Considering the cost of fuel will change once in every 24 hours, please consider introducing caching to cache cost of fuel for a requested city if not already requested within 24 hours.

Architectures used

Microservices - is an architectural style that structures an application as a collection of services that are

- Highly maintainable and testable
- Organized around business capabilities
- Owned by a small team
- Loosely coupled
- Independently deploy able

It enables the rapid, frequent and reliable delivery of large, complex applications. It also enables an organization to evolve its technology stack.

Each service is a separate code base, which can be managed by a small development team. Services can be deployed independently. A team can update an existing service without rebuilding and redeploying the entire application. Services are responsible for persisting their own data or external state.

This differs from the traditional model, where a separate data layer handles data persistence. Services communicate with each other by using well-defined APIs. Internal implementation details of each service are hidden from other services. Services don't need to share the same technology stack, libraries, or frameworks.

Event-Driven Microservices

Event-driven communication is important when propagating changes across several microservices and their related domain models. This means that when changes occur, we need some way to coordinate changes across the different models. This ensures reliable communication as well as loose coupling between microservices.

There are multiple patterns to achieve event-driven architecture. One of the common and popular patterns is the messaging pattern. It is extremely scalable, flexible, and guarantees delivery of messages. There are several tools that can be used for messaging pattern. The messaging system takes the responsibility of transferring data from one service to another, so the services can focus on what data they need to share but not worry about how to share it.

Approach to the Problem

Our Approach

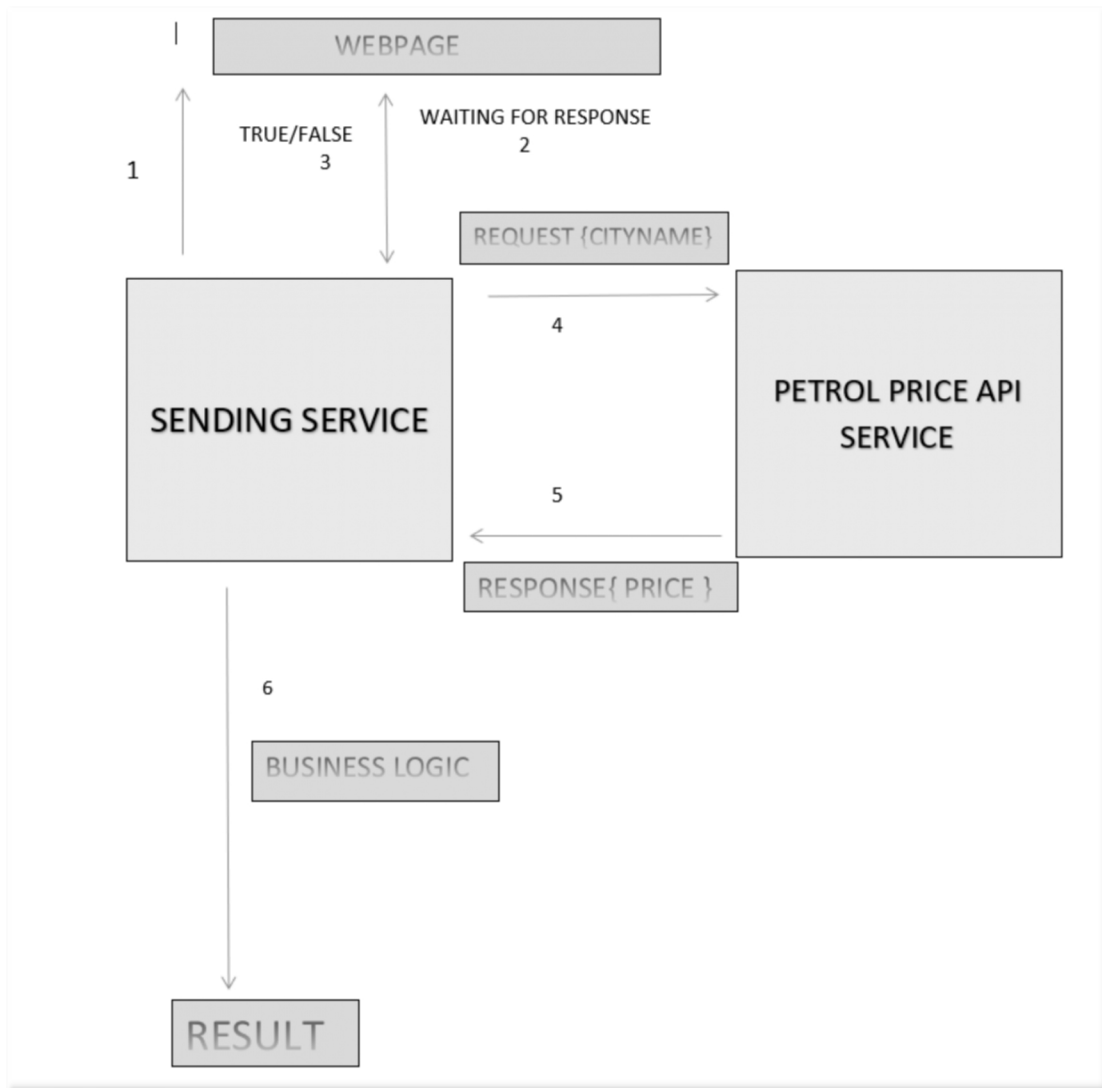
First we are creating two microservices

which are communicating with each other .When sending service is up loaded URL on web page will wait for service to respond (i.e., true/false) . If we get true as response then sending service will send request(with city name) to the other service that is petrol price API service in this case.

Then the second microservice is sending back the petrol price based on the City provided as path variable if found.

After receiving the Petrol price from the petrol price API service,first service will perform business logic and provide the required output that is the bill amount and time required to fill the tank as per the city.

FLOWCHART



Conclusion

It was a wonderful opportunity as a fresher to work on such problems. Also I got learning experience and get to know many things which includes new technology and new approach to solve problems and to understand the microservices inter communication.

With the assignment I get to know the idea of working within the time constrain environment .I tried my best to reach near about the solution if not exactly .

There are so many things to unlearn and relearn in this journey and I am all ready to do so more over I am grateful for this work.