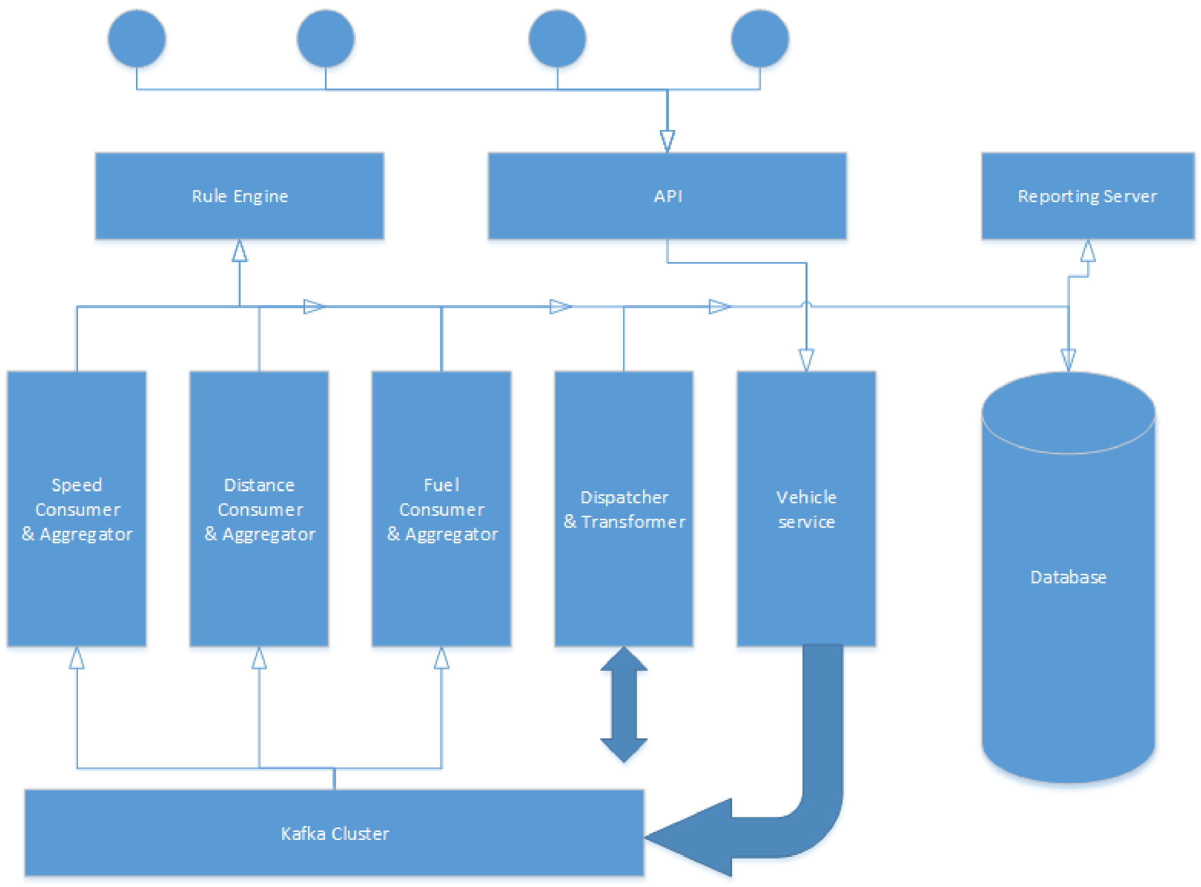
Introduction

This is description of a backend platform that is capable of receiving data from vehicles registered with the services. It exposes a bunch of APIs for vehicle registration and publishing of data. It uses Kafka for message exchange. The overall design looks as below



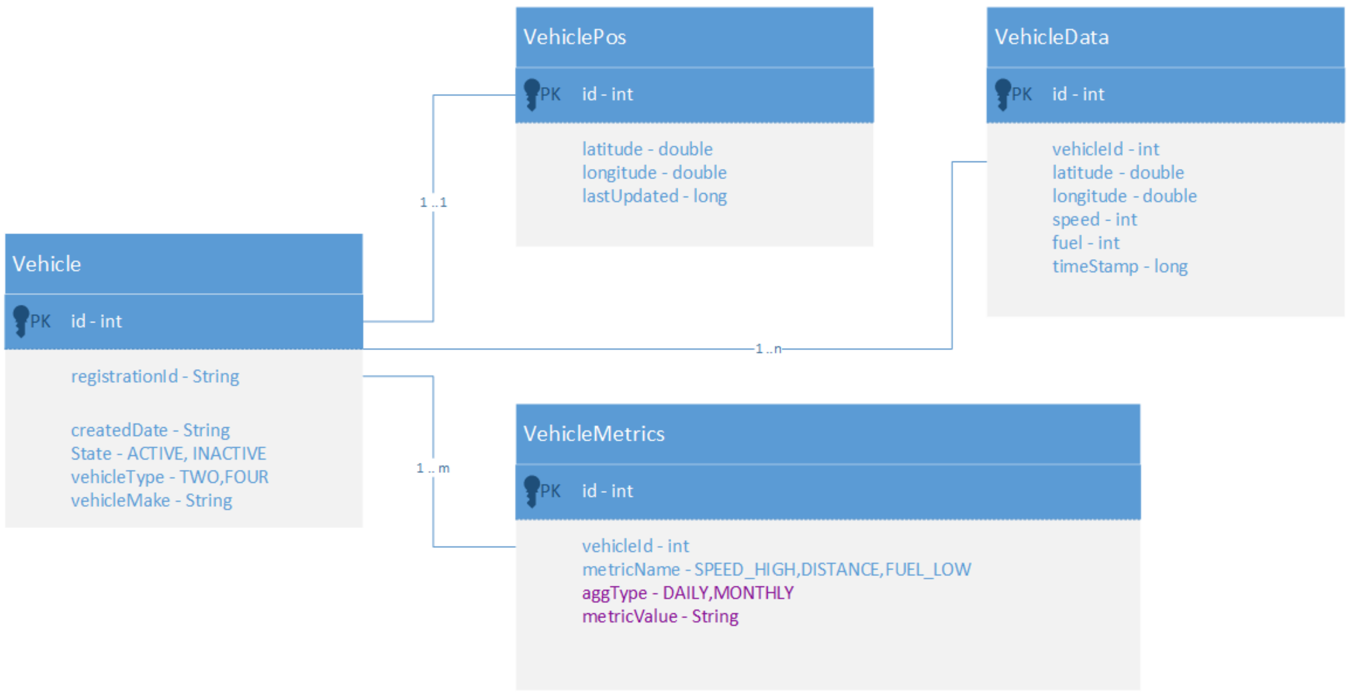
Assumptions

1. Vehicles are capable of sending their location, speed and fuel information at an periodic interval of 5 mins, either themselves or via a proxy.
2. Vehicles will be registering themselves with the backend service and will be allocated an unique id. They should be capable of sending this id along with the data.
3. Haversine formula is used to calculate the distance travelled based on the latitude. It is assumed to be relatively accurate to calculate short distances.

APIs

Method	Url	Payload
Post	http://localhost:8080/api/vmf/v1/vehicle	{ "registrationId":"ckm8812", "vehicleType":"TWO", "makeType":"Scooter" }
Post	http://localhost:8080/api/vmf/v1/vehicle/{id}/metric	{ "id": 1, "latitude":30.741482, "longitude":76.768066, "fuel": 50, "speed": 150, "timeStamp": 1677334746000 }
Get	http://localhost:8080/api/vmf/v1/vehicle	NA
Get	http://localhost:8080/api/vmf/v1/vehicle/{id}/metric	NA

Data Model



Technologies Used

Springboot, Java11, Kafka, H2 database, REST, Jasper reports, Jeasy rule engine

Scale Calculations

A typical request takes about 10ms until the metric is pushed into Kafka.

Optimum threadpool size in embedded Tomcat – 100

Total metrics that can be published to kafka in 1 second – 100 * 100 = 10000 rps

It is taking about 43 ms for data aggregation and rule verification

For each metric -> 14.33 ms

This gives an rps of 70 rps.

Few ways that this can be improved –

1. Use of cache, this can bring down the time taken for aggregation considerably.
2. Use of Kafka Streams, since Kafka is write optimized we can achieve higher rps.

Storage requirements –
Assuming a 5 mins interval in sending data, in 24 hrs a vehicle can send 288 times.
A single metric is 150 bytes. 288 * 150 – 43KB
In a month ~ 1.3Mb per vehicle. For 10k vehicles = 13GB

References

https://en.wikipedia.org/wiki/Haversine_formula

https://github.com/j-easy/easy-rules