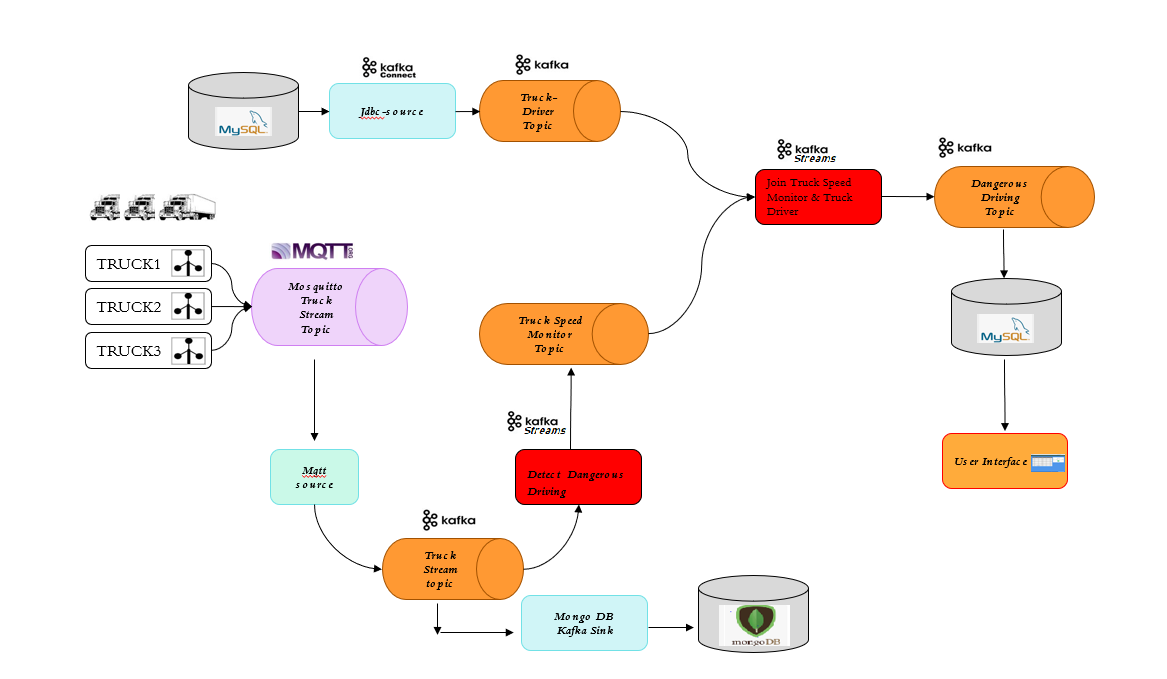
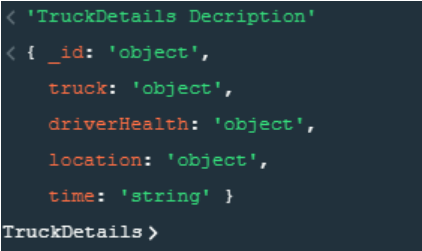
**Architectural Diagram of the Solution**



Below are the descriptions of each of the major components that we have used for designing the solution of the given problem statement.

1. **MQTT:** MQTT is widely used ISO standard client server messaging protocol. The protocol is lightweight and implements publish / subscribe communication pattern. MQTT is stable in unreliable environments of high latency and low network bandwidth. **Mosquitto** is one of the many implementations of MQTT client libraries and brokers and its purpose is to connect millions of IOT devices. But MQTT is not intended to use for high scalability and long storage of messages, or integration to business on-premise or cloud data centers. For this we need to use **Kafka** in our design.
2. **KafKa Topic:** Kafka topic follows a same publish / subscribe pattern which structures communication into topics similar to **MQTT.** Kafka is designed to be deployable as a cluster of multiple nodes which makes it excellent for scaling. It offers persistent storage of messages and integration to business on-premise or cloud data centers and applications. Main usefulness of Kafka is in distributed event streaming and store massive amount of data as messages. This makes Kafka as perfect match for the solution we are providing for the assigned problem statement.

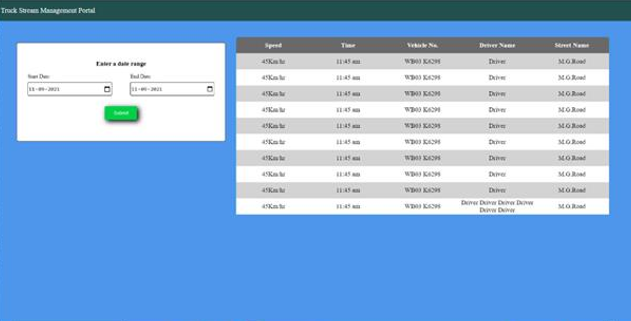
In our proposed solution we have used both MQTT and Kafka because they are a perfect combination for end to end IoT integration. We are sending the positional and speed related information of various trucks via the pre-installed IoT devices (In our solution we have used Java class to generate the streaming data for the trucks) which can connect to the MQTT broker via MQTT protocol. The messages from MQTT are then forwarded to Kafka topic known as ‘**TRUCK STREAM TOPIC**’ to process that information and help in identifying the over speeding trucks. Data from Truck Stream Topic are also stored in MongoDB using MongoDB Kafka Sink. Below is the structure of the Mongo DB collection that we have used to store the raw data from TRUCK STREAM TOPIC.



1. **Mongo DB Kafka Sink:** It is a basic Apache Kafka Connect SinkConnector that consumes data from Truck Stream Topic and saves the data into MongoDB collections.
2. **Kafka Stream:** Kafka stream is an API for writing applications to transform and enrich data in Apache Kafka, usually by publishing the transformed data on to a new topic. The data processing happens within the applications. Here we have considered the speed of 100 KPH as cutoff value. Any stream of information having vehicle speed value more than 100 KPH will be considered has high speeding vehicle and will be part of **TRUCK SPEED MONITOR** topic.
3. **DB Schema for Truck Driver Data Storage:**  The details of the drivers who are driving the truck are stored in a **MYSQL DB.** Below is the structure of the Driver table.

|  |  |  |  |
| --- | --- | --- | --- |
| **DRIVER** | | | |
| **Column Name** | **Data Type** | **Null** | **Key** |
| ID | INTEGER | N | PRIMARY |
| FIRSTNAME | VARCHAR(50) | Y |  |
| LASTNAME | VARCHAR(50) | Y |  |
| AVAILABLE | VARCHAR(2) | Y |  |
| BIRTHDATE | DATETIME | Y |  |
| TRUCKNUMBER | VARCHAR(50) | Y |  |
| CREATED\_ON | DATETIME | Y |  |
| UPDATED\_ON | DATETIME | Y |  |

1. **Kafka Connect:** Kafka connect is a tool that is reliably used for streaming data between Kafka and any other Data Bases. For the proposed solution of the assigned problem statement, we have stored the driver related information in a **MYSQL DB.** We are using Kafka Connect to stream the data between Kafka topic named **Truck Driver Topic** and MYSQL DB.
2. **User Interface:** The data in the Truck Driver Topic and Truck Speed Monitor topic are joined using Kafka Stream and stored in **Dangerous Driving Topic.** From this topic the data is Stored in a MYSQL DB. From the MYSQLDB the data is shown to the end user using an UI application. Below is the sample screen shot of the end user UI application to view the over speeding trucks.



The schema design that is being used to store the data from Dangerous Driving Topic is as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| **TRUCK\_DRIVER** | | | |
| **Column Name** | **Data Type** | **Null** | **Key** |
| ID | INTEGER | N | PRIMARY |
| DRIVER\_FIRSTNAME | VARCHAR(255) | Y |  |
| DRIVER\_LASTNAME | VARCHAR(255) | Y |  |
| ROAD\_NAME | VARCHAR(255) | Y |  |
| SPEED | DOUBLE | Y |  |
| TRUCKNUMBER | VARCHAR(255) | Y |  |
| CREATED\_ON | DATETIME | Y |  |
| UPDATED\_ON | DATETIME | Y |  |