CSC 791 / ECE 792: Internet of Things

Smart Car Fleet Monitoring System

Project Report

Abhijit Kulkarni (amkulkar)
Rashmi Sandilya (rsandil)
Purna Mani Kumar Ghantasala (pghanta)

Table of Contents

Business Objective:

Design a smart car fleet monitoring system for improved passenger safety, convenience and customer satisfaction.

1. Business requirements:

1. To get the immediate medical attention for the passengers by medical authorities and law enforcement authorities in the case of accidents.

This requirement is important for the passenger safety. In case of accident, it is difficult to receive immediate medical attention for victims if the accident occurs in remote area. The system maintains the information of the nearest hospitals in the geographical area and law enforcement authorities. This information can be used for contacting respective authorities in case of accidents so that the authorities can take actions in very quick time.

2. To enhance security of the passengers in the car during the journey.

This requirement is important for the safety of passengers from violence resorted on the passengers by the driver or other criminals. In case of situation where the driver/other criminals stop the car and act violent on passengers, the sounds generated by the passengers can be used to detect these abnormal situations and notify the law enforcement authorities. This increases the safety of the passengers.

3. To ensure drivers follow the law and order during the journey for the safety of passengers.

This requirement is required to alert the driver during the journey so that they don't violate the law and order like speed of the car. The history of the driver is maintained so that they can evaluate the performance of the driver often. This makes sure that drivers strictly follow the traffic rules and therefore safety of passengers is enhanced.

4. Optimizing route based on parameters like congestion, shortest distance and safe routes for better service.

This requirement is important to ensure that passengers are taken to the destination in stipulated time to increase the satisfaction of the customers. If there is any congestion in particular road, this information this route can be avoided to reduce the delay in traffic.

5. To evaluate the performance of the drivers in the fleet to improve the service provided by the Smart Fleet systems.

This requirement is needed to ensure that the drivers hired by the smart fleet are performing their duty without causing problems to the customers to increase the overall service of the smart fleet.

6. To improve booking system – During booking, provide the details of the cars and drivers for customer to make the best selection.

This requirement can help the passengers to choose the driver out of available cars nearby so that the system is more passenger friendly. This increases the overall passenger satisfaction.

7. Find potential consumers and market this fleet monitoring system to potential clients -Uber, Lyft etc.

2. Technical requirements:

For BR1: To get the immediate medical attention for the passengers by medical authorities and law enforcement authorities in the case of accidents.

1. Crash Sensors of 3 types - Frontal crash sensors, Sideways crash sensors, Roll-over crash sensors are deployed inside the car to detect any type of crash.
2. These sensors sense if there is any crash and in case of crash, alert notification is also sent to nearby hospitals and law enforcement authorities so that they can reach the victims immediately and provide help.
3. Information related to nearby hospitals and law enforcement authorities are sent by the data center to the fog computing unit.

For BR2: To enhance security of the passengers in the car during the journey.

1. Sound sensors are deployed inside the car to measure the noise levels continuously to observe any abnormalities like act of violence on passengers.
2. If there are any abnormalities, an alert notification is also sent to law enforcement authorities and admin staff so they can provide help to passengers.
3. Information related to nearby hospitals and law enforcement authorities are sent by the data center to the fog computing unit.

For BR3: To ensure drivers follow the law and order during the journey for the safety of passengers.

1. Speed sensors are deployed inside the car to constantly check if the speed of the car is within speed limits of the road.
2. If the speed is above limits, immediately alert notification is sent to driver to reduce the speed.
3. Updates are sent to data center related to violation of speed limit to evaluate performance of driver.

For BR4: Optimizing route based on parameters like congestion, shortest distance and safe routes for better service.

1. Dynamic route calculation based on the congestion reported by the other drivers is done by the mobile application of driver inside the car. The traffic information is sent to the application by the cloud servers whenever there is new data available at the cloud.
2. Software for best route computing is present in the mobile application.

For BR5: To evaluate the performance of the drivers in the fleet to improve the service provided by the Smart Fleet systems.

1. Feedback system to increase word of mouth from satisfied customers
2. Storage and processing of feedback.
3. Business intelligence software to monitor trends, problems and evaluate performance of the driver.

For BR6: To improve booking system – During booking, provide the details of the cars and drivers for customer to make the best selection.

1. Design a mobile app/web interface for booking and letting customer choose the car based on driver history, driver rating and type of car before booking.
2. Notification of booking confirmation and any change in booking.
3. To store information related to driver like ratings and history.

3. Components and communities:

3.1 Components:

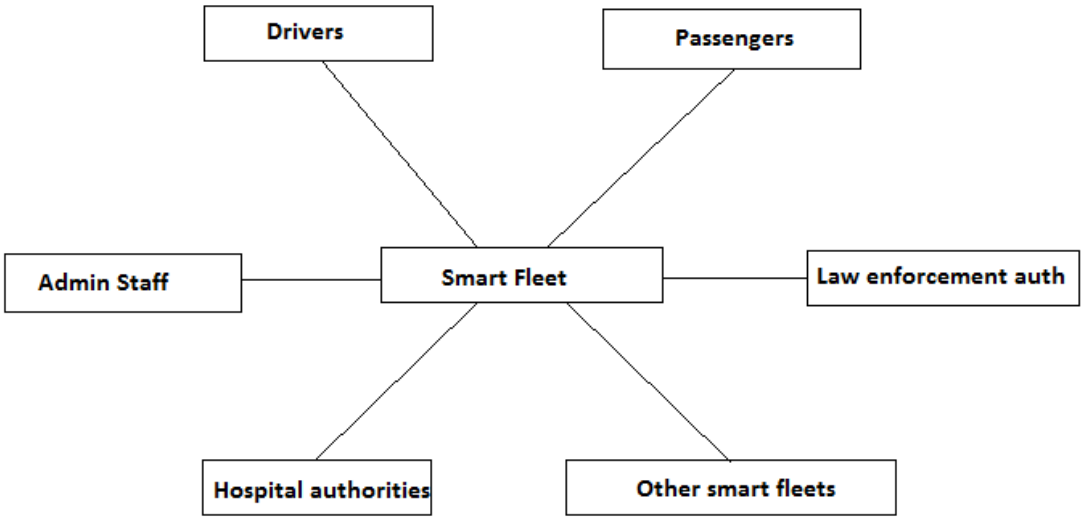
Things: Crash Sensors in Car, Sound sensor in car, Speed sensor in car, GPS Unit

People: Drivers, Passengers, Admin staff

Data: Location data, speed details, sound data, travel information, passenger details, driver details, crash/accident reporting

Processes: Booking of car, Confirmation, Pick up-drop, alert driver in case of speeding , alert driver in case of problem in location, alert medical/law enforcement authorities in case of accident/problem, computation of optimal route, analysis of traffic data, analysis of driver speed data, Billing, Customer & driver feedback

3.2 Communities:



1) IoT Admin Staff
2) Drivers
3) Passengers
4) Law enforcement authorities
5) Medical Personnel
6) Other smart car fleets

4. Conversations:

4.1 Things to Things:

1. Sound sensor sending sound level and frequencies of sounds to fog computing unit in the car.
2. Crash sensors sending crash information in case of accident to fog computing unit in the car.
3. Speed sensor sending speed of the car continuously every 2 seconds to fog computing unit in the car.
4. GPS node sending location data to the fog computing system periodically.

4.2 Things to data:

1. Fog computing system to Datacenter regarding the speed limit violation / law enforcement violation by the drivers.
2. Data sent by fog computing system to data center when there is act of violence by drivers / criminals on the passengers, accidents during journey.

4.3 Things to people:

1. Alert notification sent by the computing unit in the car to the law enforcement personnel, admin staff and hospital authorities in case of accident.
2. Alert notification sent by the computing unit in the car to the admin staff in case of assault / violence.
3. Fog computing unit in the car (based on data from speed sensor and sound sensor) sends notification to driver on mobile app.

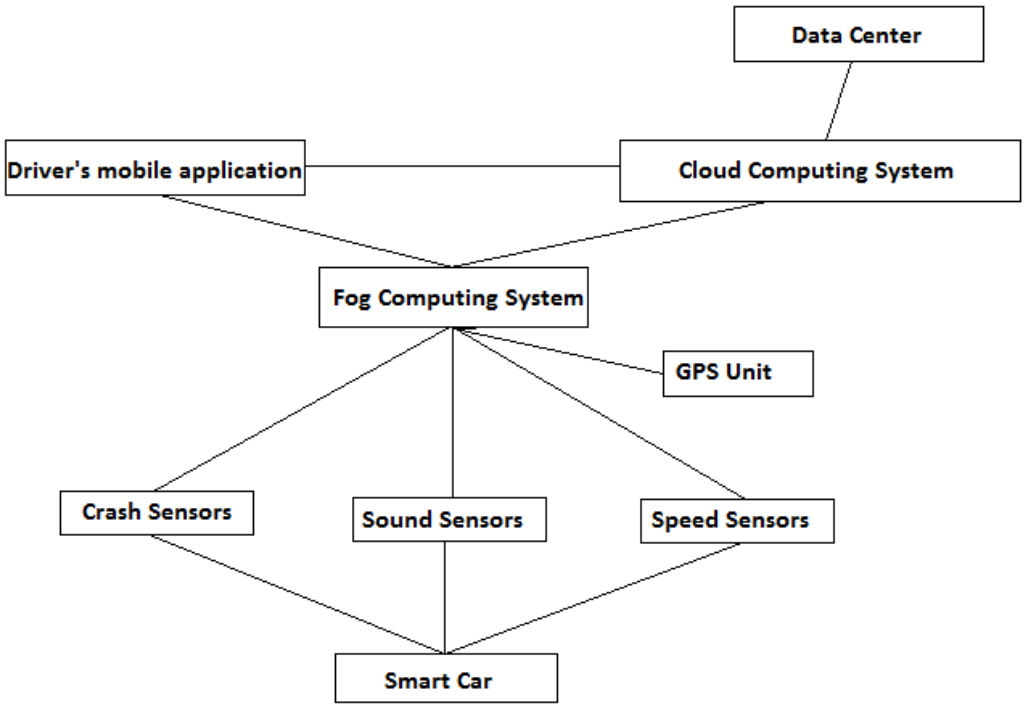
4.4 People to Data:

1. Passengers providing feedback on the travel and driver using mobile application. This information is stored in the database.
2. Hospital authorities, Law enforcement authorities providing information of the list of hospitals, information of the law enforcement authorities respectively to the Smart fleet system.
3. Drivers sending information regarding the traffic in particular route in case of congestion to the data center using mobile application.

4.5 People to People:

1. Passengers giving information of destination, route preference etc. to the driver. Driver can also ask for clarifications if necessary.

5. Architecture:



General view of the system

IoT-A architecture model applied to the smart car fleet system:

The first major contribution of the IoT Architectural Reference Model (ARM), is the IoT Reference Model itself that set the scope for the IoT design space and that address the previously discussed architectural view and perspectives.

IoT Reference Model aims at establishing a common grounding and a common language for IoT architectures and IoT systems. It consists of the submodels:

- IoT Domain Model
- IoT Information Model
- IoT Functional Model
- IoT communication Model
- IoT security and privacy Model

The foundation of the IoT Reference Model is the IoT Domain Model, which introduces the main concepts of the Internet of Things like Devices, IoT Services and Virtual Entities (VE), and it also introduces relations between these concepts.

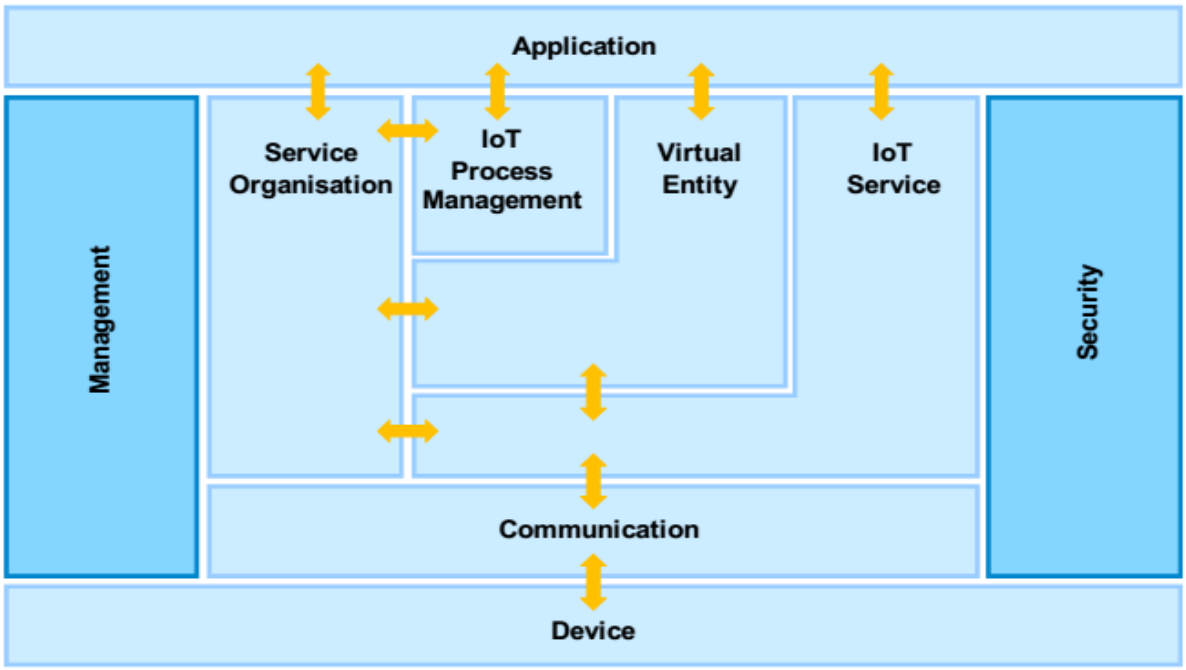
Based on the IoT Domain Model, the IoT Information Model has been developed. It defines the structure (e.g. relations, attributes) of IoT related information in an IoT system on a conceptual level without discussing how it would be represented.

The IoT Functional Model identifies groups of functionalities, of which most are grounded in key concepts of the IoT Domain Model. A number of these Functionality Groups (FG) build on each other, following the relations identified in the IoT Domain Model. The Functionality Groups provide the functionalities for interacting with the instances of these concepts or managing the information related to the concepts, e.g. information about Virtual Entities or descriptions of IoT Services. The functionalities of the FGs that manage information use the IoT Information Model as the basis for structuring their information.

The IoT Communication Model introduces concepts for handling the complexity of communication in heterogeneous IoT environments. Communication also constitutes one FG in the IoT Functional Model.

Finally, Trust, Security and Privacy (TSP) are important in typical IoT use-case scenarios. Therefore, the relevant functionalities and their interdependencies and interactions are introduced in the IoT TSP Model.

5.1 IoT-A functional model:



5.1.1. Service organization:

In general, this is split into 3 components: Service orchestration, Service composition and Service Choreography. However since the services provided in the Smart Car are modular in nature and do not inter-mix, the Service composition component is left out.

Service orchestration: The Service Orchestration FC does the resolution of the IoT Services to suitably fulfill service requests coming from the Process Execution FC or from Users.

Services offered by the Smart Car fleet monitoring system are location monitoring, speed monitoring, sound monitoring and crash monitoring. These services are used by the processes as described in the IoT process management execution functional component.

Service choreography: The Service Choreography FC offers a broker that handles Publish/Subscribe communication between services.

The speed monitoring component uses the location details provided by the location functional component. This is to determine if the speed is over the speed limit at that location/road.

The crash detection and reporting component uses the location details provided by the location functional component.

5.1.2. IoT process management:

(i) Smart car fleet process modeling : The tools that help us in the smart car fleet monitoring are speed sensors ,sound sensors, crash sensors, GPS units, mobile device of the driver (for receiving alerts), mobile device of the passenger (for booking, feedback etc.). The sensor nodes will use EDGE/UMTS/LTE for transmitting data. Fog computing is done within the car itself and data center is used for cloud computing.

(ii) Smart car fleet process execution:
These are the processes being executed within the smart car fleet monitoring system. Consider the processes for a single car:

a. Location tracking: Location of the car is tracked via GPS. The location data is sent periodically to the fog computing unit in the car and also to the data center (at bigger intervals). The data center already has the data fed in about the problematic locations and that data is transmitted to the fog computing unit in the car periodically. If there is something wrong/problematic about the current location or nearby location of the car, an alert can be automatically sent to the driver on his mobile app of the driver from the fog computing unit in the car.
b. Sound tracking: Sounds within or just outside the car can be sensed using sound sensor. If there are any abnormalities, the fog computing unit in the car sends the alert notification to law enforcement authorities in that geographical area and to the data center.
c. Speed tracking: The car speed is sensed using a speed sensor. The fog computing unit inside the car checks the speed against speed limit. If the car is above the speed limit on

that road, then an alert is sent out to the driver on his mobile. The summary and data when the car speed was above the speed limit is sent to the data center.

d. Crash detection and reporting: If there is an accident, the air bag opens up and then the crash sensor is activated. Then the location is obtained using the GPS. The nearest health care center/hospital is obtained using the location co-ordinates and the database of the medical authorities in the computing unit in the car. The information regarding the crash is then relayed to the medical authorities, law enforcement authorities and the data center.

5.1.3. Virtual entity:

These are the main components of the virtual entity functionality group:

a. Gateways (Virtual Resolution): The gateways resolve the data coming from the sensors based on the type of the data and route/forward them appropriately.
b. IPv6 address auto configuration (Virtual Entity and IoT Service Monitoring): For all the sensors and controlling/monitoring devices, the addressing is through the use of IPv6 addresses. IPv6 auto configuration is used to assign address to any new node that is connected to the IoT Network.

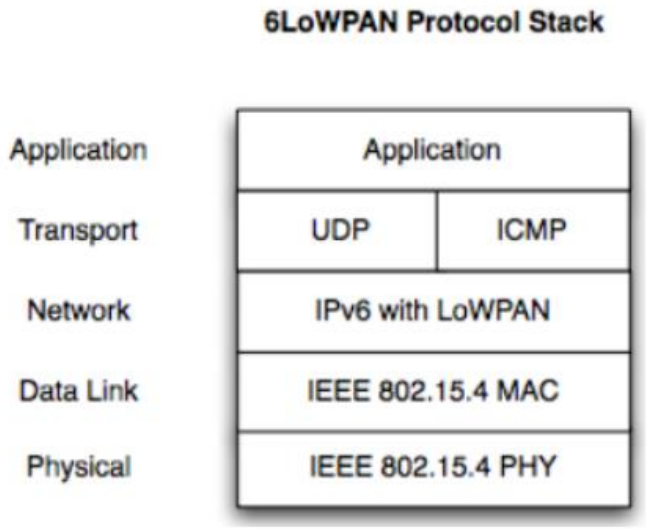
5.1.4. Communication:

Between the sensor nodes and the centers, the communication is done through wireless communication. Within the center and between the centers, the communication is through high speed wired intranet and internet. The technologies/protocols used at the various layers are described below.

Link layer: For the unconstrained devices in the system, the nodes in the data center , we will be using ethernet at link layer.
For communication between the modem in the car and the data center, we will be using EDGE/UMTS/LTE.

For communication between the sensors and the modem in the car, we are using 6LoWPAN over IEEE 802.15.4. The reason for this choice is as described below. The sensors need to be of low cost and have low battery consumption. IEEE 802.15.4 devices are designed for low cost and low battery consumption.

The full form of 6LoWPAN is IPv6 over Low power Wireless Personal Area Networks. The 6LoWPAN concept originated from the idea that "the Internet Protocol could and should be applied even to the smallest devices," and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things. The 6LoWPAN protocol defines encapsulation and header compression mechanisms that allow IPv6 packets to be sent and received over IEEE 802.15.4 based networks.



IP layer:

The addressing scheme is based on IPv6 addressing. The Address assignment is done via Auto configuration using MAC address of the device.

Transport layer:
Using TCP at the transport layer.
Using source port field as identifier for packets sent from the sensors.

Application layer:

For communication between the sensors and the data center, choosing Message Queue Telemetry Transport (MQTT) as the application layer protocol for this system at the application layer. MQTT is a lightweight event and message-oriented protocol allowing asynchronous communication between devices. It helps in efficient communication across constrained networks to remote systems. The advantages of MQTT over “traditional” protocol HTTP include higher reliability, lower battery consumption and higher speed.

For communication between the nodes of the data center, using HTTPS as the application layer protocol for convenient and secure communication.

5.1.5. IoT service:

This has two functional components:

a. IoT Service resolution: This component provides all the functionalities needed by the user in order to find and be able to contact IoT Services.

Out of the functionalities offered by this functionality component, we need resolution functionality. At the gateway, the messages need to be resolved depending on the type of service they are being used for - sound sensing, location monitoring, and speed sensing, crash detection. Using the source port field in the TCP header to distinguish between the kinds of traffic when the data from the sensors arrives at the gateway at the data center (the destination addresses of all of them will be the same)

b. IoT service: 4 IoT services are provided here:
(i) Location tracking: This is done by use of GPS in the mobile device of the driver. Helps monitor the position of a given car in the fleet at any instant
(ii) Sound monitoring: The sounds in the car and nearby are monitored using sound sensors. Of particular interest is sound monitoring for any abnormalities, ex. accidents, violence etc.
(iii) Speed tracking: This service is for tracking the speed of the car and also sending an alert to the driver automatically on his mobile app if driver is above the speed limit in that location/road.
(iv) Crash detection and reporting: This is for reporting a crash/accident by using the opening of the airbag as a signal for the crash sensor. The nearest health care center/hospital is obtained using the location co-ordinates obtained via GPS and the database of the medical authorities in the computing unit in the car. The information regarding the crash is then relayed to the medical authorities, law enforcement authorities and the data center.

5.1.6. Device:

The following devices are used:

a. Speed sensor: To monitor the speed of the car and send alert to driver if he is speeding.
b. Sound sensor: To monitor the sounds inside and near the car, for abnormalities
c. Crash sensor: if there has been accident/crash, air bag opens up and the crash sensor gets activated and sends information to the data center, law enforcement authorities and medical authorities.
d. GPS unit: It is used for GPS, to track the location of the car.
e. Mobile of driver: It is used by the driver to receive alerts - in case of speeding, possible problems near the location or on the route.

5.1.7. Application:

Mobile app is used by the driver for communication with the data center.
Mobile app/web interface is used by the customer/passenger for communication with the system.

A web interface, monitoring tools, analytics tools etc are used by the the admins of the IoT system, ex. Uber.

The IoT application along with the other tools are hosted on a cloud computing platform, like IBM Bluemix or Amazon EC2.

5.1.8 Security:

These are the components of security to be implemented.

(i) Authentication: The user/IoT customer logs into the IoT system via the internet and types in his username/password. System does not allow access without the credentials being verified.

(ii) Authorization: Not all data/functions/services of the IoT system are accessible by every logged in user. For example, someone at Uber hired to do the monitoring of the location/speed/sound sensors is not authorized to view the analytics of the driver of the car, whereas someone high up in the hierarchy like the Vice-President is authorized to view the current data as well as the analytics of the driver of the car.

The access control decisions based on access control policies set by the admin. From the design point of view, the design should be such that it allows for the setting and operation of these access control policies.

(iii) Key exchange management: For wired communication, use Public key cryptography (example: authenticated Diffie-Hellman) to establish a session key and then encrypt the rest of the conversation using that session key. For wireless communication, use WPA2 + AES to secure the communication.

Identity Management and Trust & Reputation functional components are not relevant to this system.

5.1.9 Management:

The management tasks identified are as follows:

(i) Configuration: The sensor nodes, their boards and the wireless modem within each car need to be configured to work with each other and to send data back to the centers. The IPv6 addresses for these devices are auto-configured based on their MAC addresses.

(ii) Fault: The Fault FC contains functions to handle, monitor and retrieve faults. Use a tool that monitors the data sent by a sensors -- which gives a notification if no data has been sent by a sensor for quite some time even though the car is active. In that case, there could be a problem with the sensor and it needs to be checked and fixed/replaced.

(iii) Reporting: This distills the information provided by the other functional components in the management system. This includes monitoring and analysis of driving data, Customer booking, customer feedback, sound sensor data, members of the system etc.

(iv) Member: The Member FC is responsible for the management of the membership and associated information of any relevant entity (functionality group, functional component, virtual entity, IoT Service, Device, Application, and User) to an IoT system. In the car fleet monitoring, within this functional component, the following associations are kept:

a. The association of the sensor, modem with the car
b. The association of the user with the IoT system.

(v) State: The State FC monitors and predicts state of the IoT system. Here we are not using prediction anywhere, just monitoring. The monitoring includes monitoring the state of location, sound & speed of car, crash detection, customer booking, customer feedback etc. This functionality also supports billing of the IoT Customer - depending on the use of the services by the IoT customer.

5.2 IoT-A architecture views:

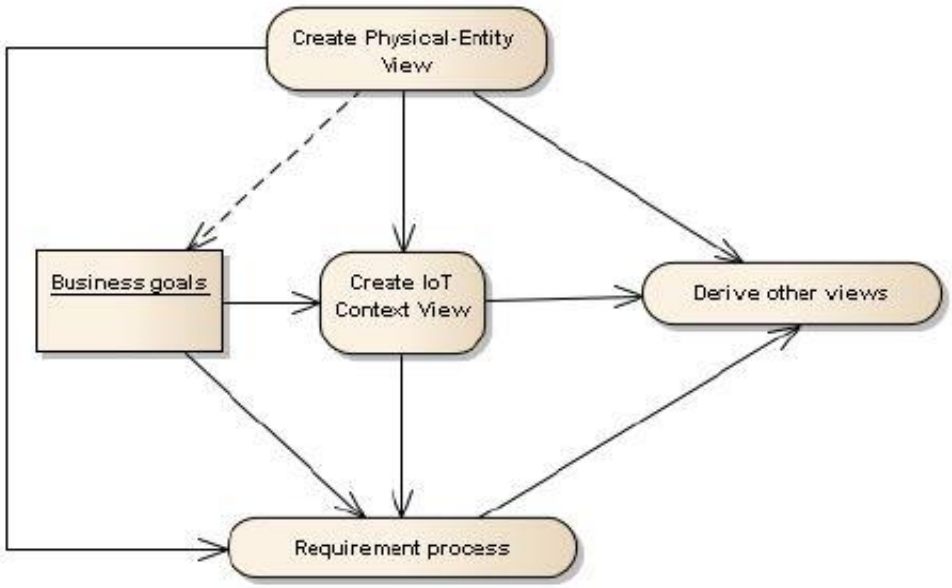
Architectural views are the main building blocks of any architecture. The system architecture depends on the views, how they are interrelated to each other and how they communicate to each other. Architectural views provide a standardized way for structuring architectural descriptions. Also segregating the architecture into different views helps in building a strong, concrete architecture and better planning.

Below are the 7 architectural views those constitute to any IoT architecture.

- Physical-Entity View
- Deployment View
- Operational View
- IoT Context View
- IoT Domain Model
- Functional View
- Information View

Generation of architecture from the views:

Below figure helps in generating architecture from different views. Also it explains which views needed to be resolved first as other views are dependent on these views.



From the figure above, it can be seen that first Physical-Entity View needs to be created and then followed by creating IoT Context View. Requirement process is dependent on both IoT context view and Business goals of the architecture. Then rest all views all are derived from the physical view, IoT context view and requirement process.

5.2.1 Physical-Entity View:

The Physical Entity is “any physical object that is relevant from a user or application perspective”. Physical entity constitutes of different sensors and what is the sensor monitoring. Also some of the sensors are placed in physical entities for various reasons.

In the IoT-A architecture for smart fleet monitoring system, different sensor devices are used for monitoring different physical quantities.

1. Sound sensor monitoring the noise levels inside the car. Noise levels are very useful for understanding the situation inside the car. Consider a situation of any violence by the car driver on the customer in the car. If the customer cannot dial 911 for assistance or cannot contact law enforcement authorities, customer will be helpless situation. In these kind of situations noise sensors which measure the noise inside the car can detect the change in noise levels. This information is sent to data center where processing is done and administrators who are monitoring these data gets the notification immediately. Further action can be taken immediately for the safety of the customer. This adds lot of security to the customers.

2. Speed sensor monitoring the speed of the car. Speed of the car should always be in the speed limit of the particular road. The speed of the car monitored is compared with the speed limit prescribed on that particular road. This information of speed limit can be received from android location based services library where the gps co-ordinates are used to get the speed limit of the road. This sensor information is processed inside the car and also in the data center.

Whenever the speed of the car is above the specified limits, the alert message is given to mobile app of the driver to bring in notice to him to reduce the speed and also this information is stored in database for calculating the performance of the driver.

3. Crash sensors are deployed to detect frontal crash, side-impact and rollover crash. These sensors measure how quickly a vehicle slows down in case of frontal crash, accelerate to side in side-impact crash and sensing system to detect rollover crash. As we know the GPS location, if the sensors detect any type of crash, with the help of fog computing in the car, the message is sent to nearest hospital for emergency assistance and also the information is given to law enforcement authorities.

4. A mobile containing IoT app is also a device because it gives gps location of the driver and it is used for calculating the location of the car periodically. Also this information is used for calculating the speed limit of the car at those coordinates and compare with the speed of the car. The GPS location of the car is also sent to datacenter so that they can track the car and store the information.

5.2.2 IoT Context View:

Context View describes “the relationships, dependencies, and interactions between the system and its environment (the people, systems, and external entities with which it interacts)”. To be more specific, the context view describes “what the system does and does not do; where the boundaries are between it and the outside world; and how the system interacts with other systems, organizations, and people across these boundaries”.

With respect to smart fleet monitoring IoT application, there are many communities involved and several interactions within communities. Context view describes these interactions between the communities and also boundaries to the communities.

Several communities are Management system providers, drivers, passengers, medical personnel, law enforcement authorities, other smart fleet monitoring organizations.

Management system providers has access to the entire data collected and sent by fog computing systems present in the car. The data stored in the data center related to statistics of driver performance (number of times the driver violated speed limit), feedback from the passengers on the driver, and past history (containing the accidents done by driver, number of times driver resorted to violence) should be accessible to only the management system providers. Drivers, passengers should be able to get this data. It is the responsibility of the management to use this data and to take actions accordingly.

Hospital authorities share their geographical location of the hospital, list of doctors, contact of the hospital to the smart fleet monitoring administrators so that this information can be used in case of accidents are detected by sensor so that nearest hospital authorities are notified and emergency actions like sending ambulance can be taken by the hospitals.

Law enforcement authorities share their details to smart fleet monitoring authorities and this case information can be used in case of accidents so that they can come into the accident location immediately just like discussed above with hospital authorities. Also in case where driver resorts to violence against passenger, sound sensors sense the abnormalities and fog computation system sends the message to law enforcement authorities so that they can rescue the passenger in prompt time.

Drivers get the data of his own performance in regular intervals of around 1 to 2 days so that he can be cautious.

Traffic information recorded by one smart fleet monitoring organization can be given to other smart fleet monitoring organizations and vice-versa so that they have real time information on live traffic congestion.

This smart fleet monitoring application can improve the customer satisfaction and safety of the travel by the above sensors and information from the sensors combined with different communities. This cannot guarantee that accidents can be avoided, driver resorting to violence on the passenger, transporting the passengers the destination in specified time.

5.2.3 IoT Domain Model:

IoT Domain Model provides a semantic and ontological overlay for the context view in which it provides guidance on the core entities of an IoT system and how the entities relate to each other. It also aids in identifying system boundaries, which is one of the main questions to be addressed in the context view. IoT domain model helps identifying Devices, identifying Resources, the Services to be used - where they should be deployed are analyzed and finally the Users of these Services are identified.

The devices are sensors mentioned in the physical entity view of the architecture.

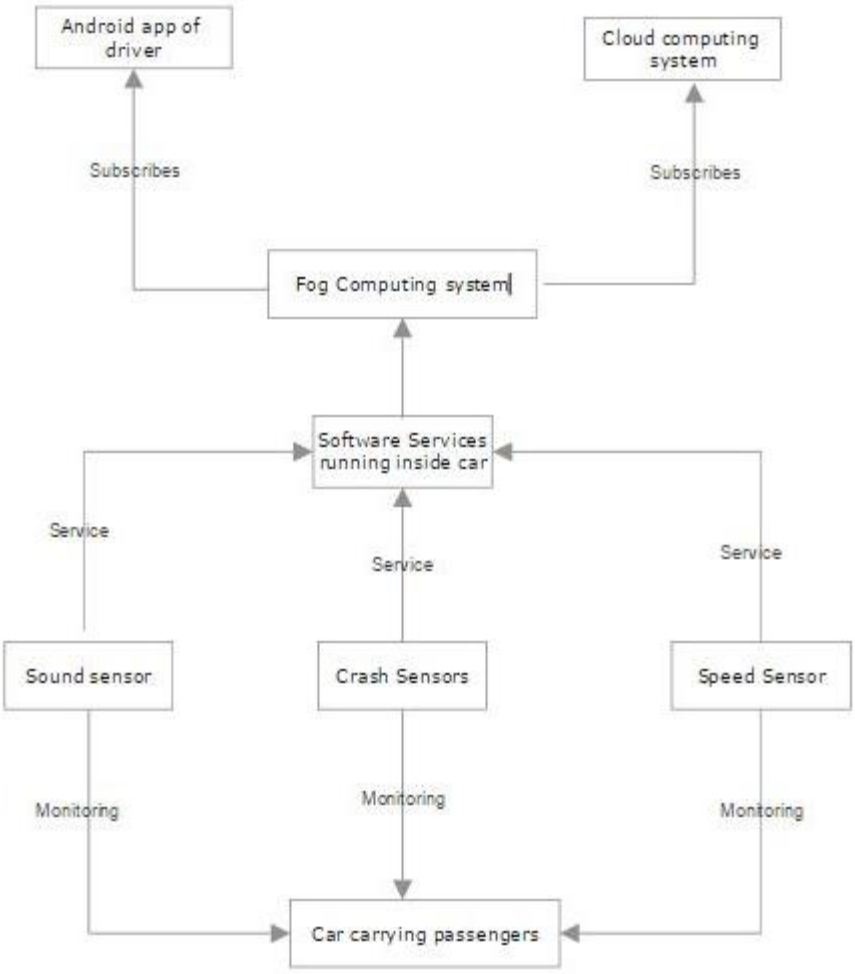
Resources are software components that provide information about or enable the actuation on Physical Entities. In the case of our smart fleet monitoring system, mobile application carried by the user can be used as actuator. Consider an example where driver is above the speed limit of particular road. Fog computation is done inside the car on the speed of the car and information is sent to mobile application so that it can alert the driver that he is above the speed limit. Also the different network components that enable the functionalities of different components will come under the resources. In the cases of crash sensors and sound sensors, the software runs directly on the device to monitor the quantities those are measured and send to the computation model. These softwares providing the information from the sensors are also part of Resources. The software that is used in the hospitals for providing the information of the hospitals, software in law enforcement authorities for providing law enforcement authorities details also comes under resources.

A Service provides a well-defined and standardized interface, offering all necessary functionalities for interacting with Physical Entities and related processes. Often it exposes a functionality provided by a Resource to the overall IoT system. In the mobile smart fleet monitoring application, we can distinguish different types of services running in the car and also in the cloud.

The software inside the car keeps on checking the speed of the car and if the speed is more than the limit, alert is send to the driver through mobile app. This software which keeps on checking the speed can be considered as service. Also noise monitoring using sound sensors are sent to the local computing device which contains software and checks again the threshold noise levels. Whenever abnormalities are observed, then the information is sent to cloud computing and also alerts the law enforcement authorities. This software provides the functionality of the interacting physical entities. Crash sensor data is also fed to the software running to check if there is crash. These all are the services which provides

A User interacts with a Physical Entity, physically or mediated through the IoT system. In the case of a mediated interaction, a User invokes or subscribes to a Service. In Smart fleet monitoring system, Drivers, Administrators, Customers all comes in the User. Administrators subscribes for the alerts regarding abnormal situations and also the drivers. Passengers interacts with the system as they are the one’s using the vehicles for travelling and providing feedback about the drivers.

IoT Domain Model figure can be shown as below:



5.2.4 Information View:

The Information View shows how the information flow is routed through the system and what requests are needed to query for or to subscribe to information offered by certain functional components. Information view provides what physical quantities are monitored by the sensors and how are the quantities related to each other. One of the main purposes of connected and smart objects in the IoT is the exchange of information between each other and also with external systems. Therefore the way to define, structure, store, process, manage and exchange information is very important. The information view helps to generate an overview about static information structure and dynamic information flow.

Information View focuses on the description, the handling and the life cycle of the information and the flow of information through the system and the components involved. Information view provide a viewpoint for modelling the type system of virtual entities.

Information description covers identifying the virtual entity, give the service description

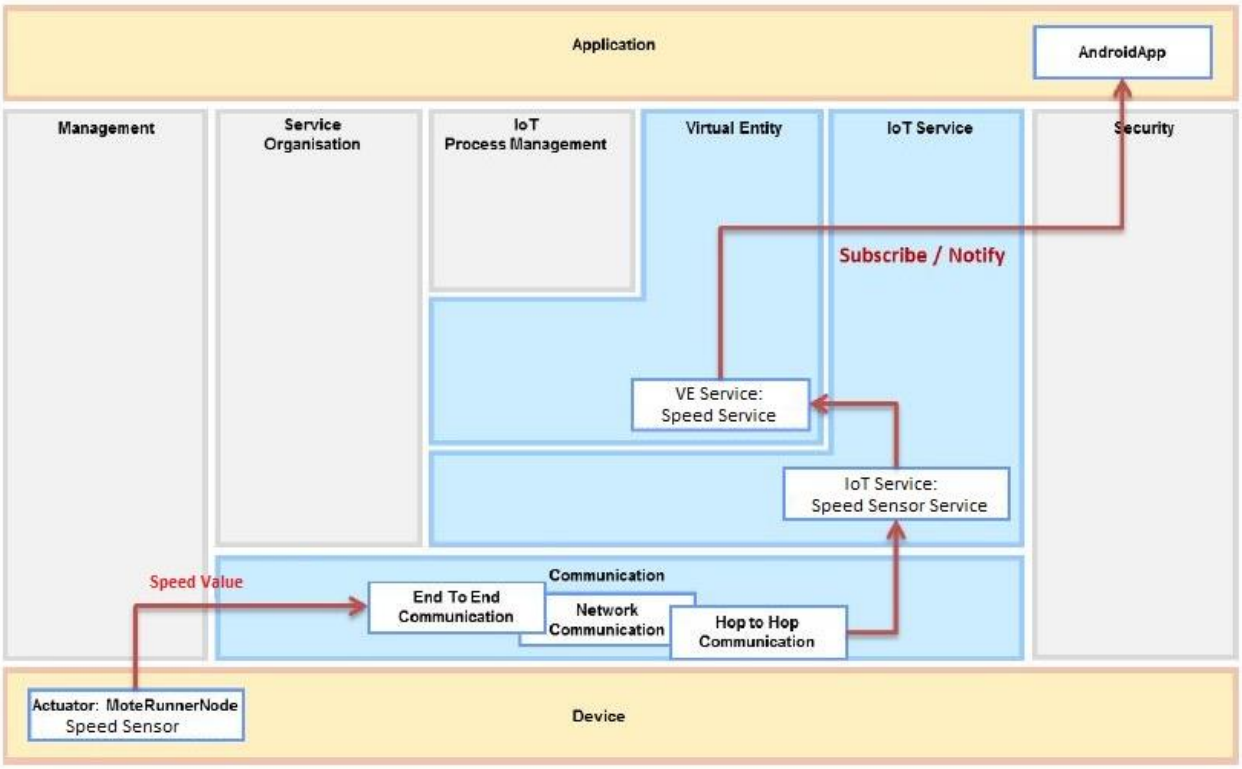
Virtual Entity: The Virtual Entity models the Physical Entity or the Thing that is the real element of interest. Virtual Entities have an identifier (ID), an entity Type and a number of attributes that provide information about the entity
For aspects of smart fleet monitoring system, virtual entities are Driver, passengers, administrators and different type of sensors which generates the data for fleet monitoring. The hierarchical view for the virtual entity modelling for smart fleet monitoring can be given as :

Service Description: Service Descriptions contain information about the interface of the service, both on a syntactic as well as a semantic level, e.g. the required inputs, the provided outputs or the necessary pre-conditions as well as post-conditions. Furthermore, the Service Description may include information regarding the functionality of the resources, e.g. the type of resource, the processing method or algorithm etc., or information regarding the device on which the resource is running, e.g. its hardware or its geographical location.

Association between services and virtual entity depends on the attribute of the virtual entity for which service provides the information or enables the actuation as a result of a change in its value. In smart fleet monitoring association between services and virtual entity depends on the attribute shown in hierarchical entity diagram as shown above.

Information Handling: Information in the system is handled by IoT Services. IoT Services may provide access to On-Device Resources, e.g. sensor resources, which make real-time information about the physical world accessible to the system. IoT Services are registered to the IoT system using Service Descriptions. Service Descriptions can be provided by the services themselves, by users or by special management components that want to make the service visible and discoverable within the IoT system. Associations can be registered with the VE Resolution by services that know for what Virtual Entities they can provide information. The registration can be done by users, by special management components, or by the VE & IoT Service Monitoring component. In smart feet monitoring the VE's like speed sensors generates the data. This data is transported to the local fog computation system sitting on the car and based on the computation results alert is sent to the application on the mobile device of the driver if the speed limit is beyond the permitted limit.

Diagram for information handling for smart fleet monitoring can be given as:



5.2.5 Deployment and Operational View:

Deployment and Operation view is very important to address how actual system can be realized by selecting technologies and making them communicate and operate in a comprehensive way. The Deployment and Operation View aims at providing users of the IoT Reference Model with a set of guidelines to drive them through the different design choices that they have to face while designing the actual implementation of their services. Choosing among the different connectivity types is not as straightforward as different choices may provide comparable advantages, but in different areas.

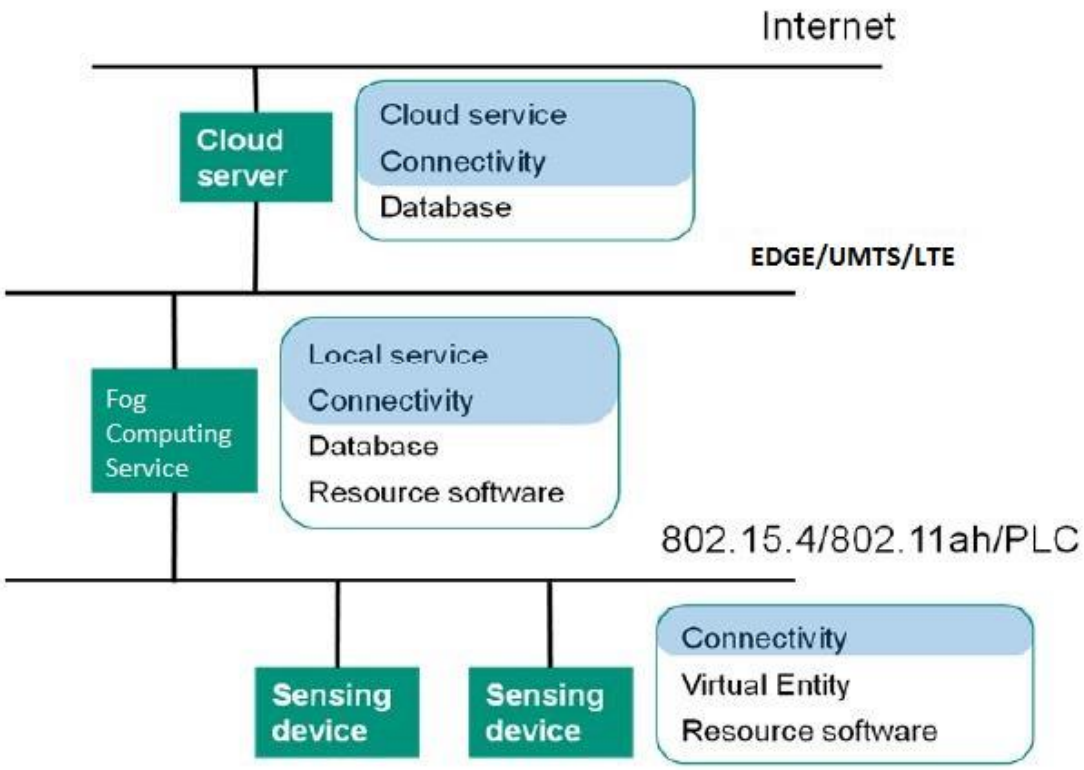
The communication between the sensors and the fog computing system in the car can be implemented using 6LoWPAN over IEEE 802.15.4. 6LoWPAN is IPv6 over Low power Wireless Personal Area Networks which can be used for low power devices with limited processing capabilities. The sensors should have low battery consumption and low cost. 6LoWPAN serves this purpose.

EDGE/UMTS/LTE can be used for communication between the fog computing system inside the car (connected to 3GPP modem) and the data center. This will provide wireless connectivity in everywhere hence this is ideal for the IoT applications which involves mobility of the devices.

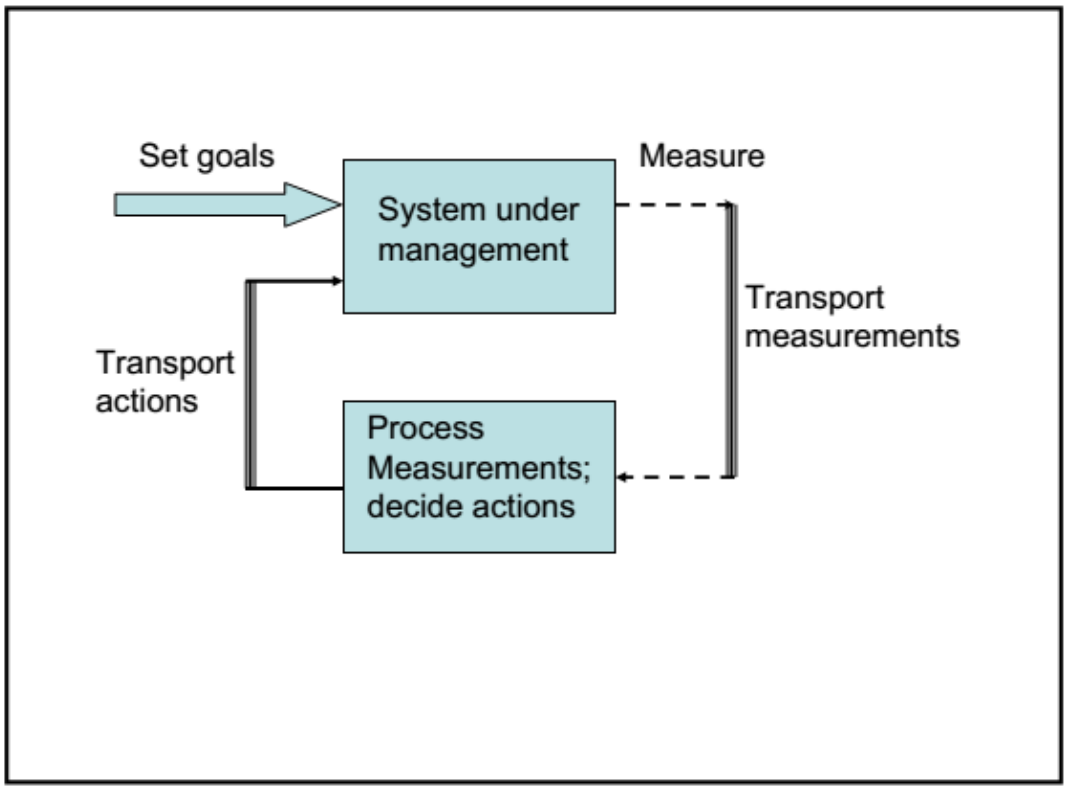
Within the datacenter and between the distributed data centers, the communication is through high speed wired intranet and internet.

Now as we have defined communication between different devices, now it is important to define where to deploy services and resources which are the softwares related to a given device. As a part of smart fleet monitoring system, software is run on all the sensors devices inside the car to measure the quantities and communicate to the edge fog computing system. Also other softwares are run on the fog computing system related to sends alert to mobile application of driver which acts as a actuator

Figure of Deployment & Operational View can be shown as below:



5.3 Feedback model:



1. Goal: Monitoring the location of the car

Measuring: The location of the car is obtained using GPS.

Transporting measurements: The location co-ordinates are sent from the GPS unit to the computing unit in the car and the modem in the car using 6LoWPAN. The location co-ordinates are periodically sent from the modem in the car to the data center via UMTS/EDGE/LTE.

Processing measurements:
The location co-ordinates are consistently compared with locations of the problematic areas that are in the computing unit. Note that the problems and the area/location is already in the IoT system database and periodic updates are sent out to the computing unit in the car.

Deciding and transporting actions: If the location co-ordinates are close to a problematic area, an alert is sent on the driver's mobile app. This is sent from computing unit to the driver's mobile using UMTS/EDGE/LTE.

2. Goal: Monitoring the speed of the car

Measuring: The speed of the car is measured using speed sensor.

Transporting measurements: The speed data is sent from the GPS unit to the computing unit in the car and the modem in the car using 6LoWPAN.The car speed is sent periodically from the modem in the car to the data center via UMTS/EDGE/LTE.

Processing measurements: The computing center within the car gets the present speed of the car via the car sensor, compares with the speed limit on that road/location. The location data obtained from the GPS and a location based library is used to get the speed limit at that location.

Deciding and transport actions: If the speed of the car is more than the speed limit at that location, an alert is sent to the driver on his mobile app. This is sent from the computing unit to the driver's mobile using UMTS/EDGE/LTE.

3. Goal: Monitoring car for any sounds that indicate possible problems in the car (accident, violence etc.)

Measuring: The sounds are measuring using sound sensor.

Transporting measurements: The sound measurements are sent from the sound sensor to the modem within the car using 6LowPAN. Then sent from the modem to the data center and law enforcement authorities using EDGE/LTE/UMTS.

Processing measurements: If the sound frequency is abnormally high (ex. a gunshot) or very high for a prolonged period of time (say 15-20 seconds), that may indicate a possible problem in the car. The measurements are processed to check for these within the computing unit in the car.

Deciding and transport actions: Based on the above processing of the sound measurements, an alert/notification is sent out to the data center and to law enforcement authorities if it matches the criteria mentioned. This is sent from the modem in the car to the data center and law enforcement authorities using EDGE/UMTS/LTE.

4. Goal: Monitoring for car crash/accident

Measuring: If the air-bag opens up, the crash sensor is activated.

Transporting measurements: The crash sensor activation is sent to the computing unit within the car using 6LoWPAN.

Processing measurements: When the crash sensor is activated, the location data is obtained from the GPS unit to relay the location of the car. This is done at the computing unit within the car.

Deciding and transport actions: Alert is sent out to the data center, medical authorities and law enforcement authorities with the car details and location details.

5.4 Comparing IoT-A model with feedback model:

1. IoT-A model provides different views of the system with the audience in mind. For example, the deployment and operation views described here will help the administrators of the system in carrying out their management tasks - e.g. which computing is to be done via fog computing and which computing is to be done at the cloud. The context view describes “what the system does and does not do; where the boundaries are between it and the outside world; and how the system interacts with other systems, organizations, and people across these boundaries”. The feedback model does not present these different views.

2. The feedback model is easy to apply as it involves simple steps for every goal for the IoT system. In the smart fleet monitoring IoT, we have 4 such goals. The IoT-A model is tougher to apply in comparison as it is a detailed process consisting of many views and models.

3. The IoT-A model helps us in getting a complete picture of the system. For example, it clearly specifies what happens in the car with the various sensors, computing unit, modem, driver, passenger etc., what happens in the data center, how the data is transported etc.The feedback model does not give a complete picture of the system. For example it does specify how the admin can carry out the management tasks, it does not describe the security of the system components, of the data transported within the system etc.

4. The IoT-A model specifies how to identify things, data etc. as part of views and models. For example, here data from sensors is identified on the basis of TCP port# at the gateway of the data center. Identification of things like sensors, modems in the system are clearly specified in the IoT-A model. The modem in the car is identified on the basis of its unique global IPv6 address, the sensor is identified using the global IPv6 address and the TCP source port# in the data packet. The feedback model does not specify how to identify things, data etc.

5.5 Answers to the 8 basic questions on the basis of the architecture:

How do we identify things?
In context of smart fleet monitoring system, things are different objects on which sensors are attached for example objects with speed sensors, objects with crash sensors, objects with

sound sensors etc. These objects are IoT devices which can be identified based on IPv6 address and device ID (a field in the layer 4 of IoT architecture. Note: Device ID can be assumed to have similar concept as we have for port numbers in internet architecture). All the objects with sensors present in a particular car will have same IPv6 address but can be distinguished based on their device ID. The similarity can be drawn from Internet architecture where different type of application like http, SSL etc uses same IP address but are distinguished based on their port numbers.
Note that alternatively, in sphere of IoT system, things can also be uniquely identified based on "smart" RFID tags which combines sensors with RFID tags. These RFID tags can be passive tags with external input. One such RFID chip can be "MCRF 202" RFID chip with 1 bit sensor input. Again we need to use RFID tags combined with IP address to identify things in IoT sphere.

How do we identify data, people and processes?

Data from the same node is identified based on the flow label in IoT sphere. In case of smart fleet monitoring system, data from different sensors let's say speed and location can be distinguished based on identification of the device which can be done by modifying the Layer 4 say putting a field in layer 4 header instead of port number which will contain information about which device is generating the data. The sequence of data generated from the same device can be identified based on flow label.
The people in smart fleet monitoring systems are Drivers, passengers and admin staff. Each of these people will be identified based on unique authentication i.e username and password. Each user will be having unique credentials to log in to the system to validate the identity.

Processes in smart fleet monitoring are booking system, feedback system, safe travel from source to destination etc. These processes consists of many individual steps that have to be executed in series or in parallel. There are several "things" and "data items" associated with individual steps of each process. Process identification can be done on the basis of business requirement. Identifying the process also involves identifying the things and data items for each of these process. For example the things for the booking system can be mobile devices, cloud computing server. Data items can be "list of cars nearby", "driver history" etc.

How do we discover things?
Multicast DNS is a way of using familiar DNS programming interfaces, packet formats and operating semantics, in a small network where no conventional DNS server has been installed. Multicast DNS is a joint effort by participants of the IETF Zero Configuration Networking (zeroconf) and DNS Extensions (dnsext) working groups. The requirements are driven by the Zeroconf working group; the implementation details are a chartered work item for the DNSEXT group. Most of the people working on mDNS are active participants of both working groups. IPv6 neighbour solicitation and advertisement mechanism can also be used for discovering things in wireless sensor network. In case of smart Fleet monitoring system we can use multicast DNS and neighbour solicitation with advertisement mechanism for discovering the group of sensors (things) for each of the cars/fleets.

How do we discover data and people?
People can be discovered in smart fleet monitoring system based on their identification and authentication mechanism. For example drivers can be discovered based on their driver identity card which will be having the driver's Id. The driver's ID will also be maintained in database of the system so that administrator can discover the driver's location and other details based on this. Similarly discovering mechanism will work for people like administrators, fleet managers etc. Passengers can be discovered let's say during the trip based on the unique credentials they get when they first associate with the system. The mechanism used for discovering will be location tracking of the car in which passenger is travelling. Discovering data is more of identifying data. Data can be identified based on the flow label which will enable us to know the sequence of data.

How to connect things to things? things to data? things to people?
Things to Things: Wireless links are the most commonly mechanism used for connecting things to things. For example in case of our smart fleet monitoring different types of sensors in the car can communicate the measurements to the datacenter through these wireless link. These wireless link can use edge wireless technologies like LTE which can provide high speed data transfer. Also the wireless technologies like HSPA and UMTs can be deployed for this connection. The communication between data centers utilizes the wired layer 2 technologies like Ethernet and LAN.

Things to data: The primary value that IoT creates is a direct result of the data that can be captured from connected things — and the resulting insights that drive business and operational transformation. The connection of the end devices to cloud computing data centers for providing computation and storage of data is an example of things to data connection. In case of smart fleet monitoring system different types of sensors deployed in the car (things) connects to the cloud computing data centers for transporting the measurements utilizes things to data communication.

Things to people: Connecting things to people requires not only identification of different interfaces for things but also their reusability. This means that different interfaces should be integrated more than being user friendly. If user(fleet monitoring company personnel) needs to learn different interfaces for example managing speed data, sound data, crash sensor data and location data, then the purpose of smart monitoring using Internet of things is not solved. A single interface at the end point should be capable of managing all the data. This was connection of things to people can be made smarter.

How do we actually forward traffic among two elements of an IoT system? That Is, how do we find a "path" from one thing to another?

Forwarding traffic among two elements of an IoT system requires using routing protocol. The protocols like OSPF, BGP, RSVP and source routing can be used to determine path from one thing to another. Another mechanism that can be used for forwarding traffic is broadcasting.

In case of our smart fleet monitoring system we have fog computing based system sitting in the car so all the sensors in the car will be transporting the measurement to the fog computing based system which will do the local computation and will be updating remote cloud computing based datacenter occasionally.
The communication between the sensors and the fog computing system in the car can be implemented using 6LoWPAN over IEEE 802.15.4. 6LoWPAN is IPv6 over Low power Wireless Personal Area Networks which can be used for low power devices with limited processing capabilities. The sensors should have low battery consumption and low cost. 6LoWPAN serves this purpose.

EDGE/UMTS/LTE can be used for forwarding traffic between the fog computing system inside the car (connected to 3GPP modem) and the data center. This will provide wireless connectivity everywhere hence this is ideal for the IoT applications which involves mobility of the devices.

Within the datacenter, the traffic is forwarded through high speed wired intranet and internet.

How do we compute in IoT, especially when so much is promised?
When (lots of) sensors are connected, they produce massive information. Computing refers to generating (business) value out of this information using analytics. Analytics in IoT refers to making use of measurements/data provided by things and people to help the user of an IoT system (e.g., an enterprise) make (better) decisions.
In case of smart fleet monitoring, one of the behavioral analytics can be to analyse the driving speed of the driver. Based on the data from speed sensors over a period of time, the driver's driving skills can be determined. Another analytics in this sphere can be to analyse the type of car selected by passenger while booking. Based on this data, demand can be analysed for that particular type of car and the number of those cars can be increased in the system.

How do we manage and control things/elements of an IoT system?
Connections between, say, things allow for two-way communications. The flow of information towards a thing provides the basic framework to send commands to it and thus control it. The elements or things can be controlled in IoT system based on the concept of feedback model where the data/measurement from sensors are transported to actuators where data is processes and the relevant actions to be taken are generated. These actions are transported back to sensors in the system under management.

In smart fleet monitoring vertical, the data from the speed sensors, crash sensors, sound sensors etc are transported to the fog computing system sitting in the car locally. The actuator in the fog computing system will be processing the data and generate the actions to be taken, these actions will be transported back to system. For example speed sensors data will be

processed and action will be taken in form of alert sent to driver's mobile phone to reduce speed in case speed limit is exceeded.

6. Computation model:

Smart Fleet monitoring has both fog computing and cloud computing. Fog computing is done at the edge of the architecture i.e. in the car and all the edge nodes. Fog computing systems are connected to centralized cloud computing system.

6.1 Fog computing:

Fog Computing is a paradigm where cloud computing is extended to the edge of the network. This creates a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional cloud computing data centers. In our IoT application, fog computing is done inside the car using IBM Bluemix board. This is the fog computing unit mentioned in the document elsewhere and is attached to a modem.

(i) Speed limit checking: Location is tracked using GPS unit. The location co-ordinates are used to compute the speed limit at that location. There is a constant comparison of the car speed with the speed limit at the location. This is done at intervals of 2 seconds. This computation is done within the processing unit in the car. If the speed of the car is above limits, immediately the information is sent to the mobile app of the driver. This serves as the purpose of actuator. Also the information is sent to data center.

(ii) Accident reporting: If the crash sensor is activated as a result of an accident and the air bag opening up, then the location is obtained using the GPS. The nearest health care center/hospital is obtained using the location co-ordinates and the database of the medical authorities in the computing unit in the car and alert is sent to the health care center/hospital for emergency assistance. This computation is done within the processing unit in the car.

(iii) Sound monitoring and reporting: Sound sensors monitor the sound frequencies inside the car constantly and compared to the regular frequencies. If the abnormal frequencies which corresponds to shouting are recorded for more than specified time approximately 5 - 10 secs, this could be due to some abnormal situations inside the car and fog computing system inside the car alerts the law enforcement authorities so that they can take further actions. This computation is done within the processing unit in the car and also this information is sent to the datacenter.

(iv) Route / location related problems: Location is tracked using GPS unit. The computing unit has the data of the problematic locations and the problems. It checks with the current location of the car. If the car is nearing a problematic area, it sends a notification to the driver on his mobile app.

6.2 Cloud computing:

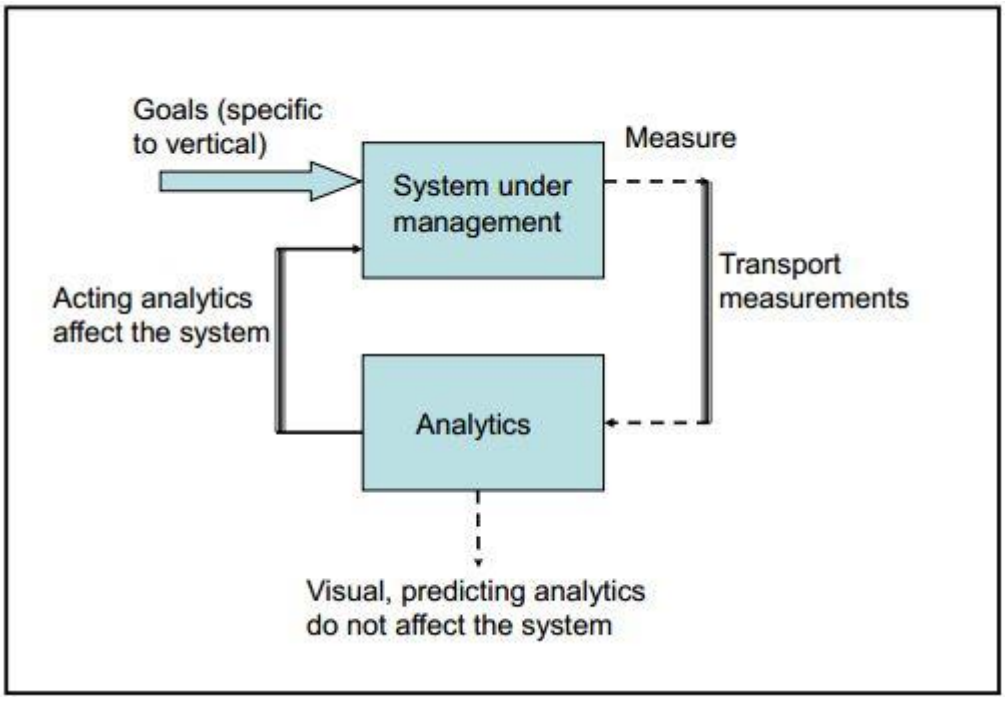
Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet). Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand.

Cloud computing is used in various steps of processes in Smart Fleet monitoring application. During initial booking of the car by the customers, mobile application used by customers sends request to centralized cloud computing server. The results sent by the server will be populated in the mobile application used by the customer. There is an option for customers to select a particular car out of many cars those are nearby. During journey, all the sensors report data to fog computing system. If there are any abnormalities as discussed in the architecture, fog computing system sends those details to the cloud computing system which are processed by the servers and stored in the database. The database contains all the details of particular driver, route used by the driver, speed regulations violated by the driver during every journey etc. The feedback sent by the customers is also processed in the cloud server and stored in the database. The servers running in the cloud also subscribe to different communities like law enforcement authorities, hospital authorities so that they can received information and updates from these communities and server updates the database with the changes published by these communities. This data is sent to the fog computing system and also mobile application used by the driver to calculate route. The administrator can run tools or softwares for fetching data from the cloud computing system to assess drivers performances, feedback upon the service and drivers.

7. Analytics:

Analytics in IoT refers to making use of measurements/data provided by things and people to help the user of an IoT system (e.g., an enterprise) make (better) decisions.

Analytics can be explained using feedback model as follows:



Analytics in IoT space can be classified according to the level of its difficulty or intelligence; a widely used classification uses three levels - *monitoring*, *predicting* and *acting* analytics. There are some of the analytics issue common to all the verticals. These issues are:

1. Data summarization
2. Data integration (from many disparate sources)
3. Trends analysis
4. Data in Motion
5. Complex event processing (CEP)
6. Resistance to change: few admins feel prepared to make big decisions (based in part on analytics findings)

Analytics in Smart fleet monitoring system:

1. Behavioural analytics to analyse the driving speed of the driver.
2. Analysis of type of car selected by passenger while booking.
3. Using predictive analytics techniques, which take an understanding of the past to predict future activities and model scenarios using predictive models, simulation and forecasting. These analytics will help in fraud detection in case of booking system like some third party promoting some fake offer for free ride using some promo code etc.
4. Fleet owners can actively explore the booking history to measure passenger's trip details, frequency of taking ride with the particular fleet etc and suggest the appropriate recommendations to target the right passengers.

5. The feedback from the passengers can be collected and analyzed to determine the driver's driving skills, the overall passenger satisfaction etc.

8. Enabling technologies:

IPv6: IPv6 is used for addressing every node in the network. This choice is for a number of reasons, the primary ones being the depletion of addresses in IPv4, the ability of a node auto-configure IPv6 address based on its MAC address.

Sensors: Sound sensors, Crash sensors, speed sensors are required for the monitoring of sound within or near the car, possible crash/accident of the car and speed of the car respectively.

GPS: GPS is required for tracking the location of the car.

Mobility: Technologies that enable transfer of data while on the move - 6LoWPAN, IEEE 802.15.4, EDGE/UMTS/LTE are used in the smart car fleet monitoring system. 6LoWPAN & IEEE 802.15.4 are required for communication between sensors and the modem/computing system within the car in a cost efficient way and with low power consumption. EDGE/UMTS/LTE are required for other wireless communications.

Virtualization: In the smart fleet monitoring system, virtualization is only at the data center. The data center physical machines are subdivided into appropriate number of virtual machines (with sufficient memory, processing power, disk space in each virtual machine). Each virtual machine has its own IP address.

Data visualization: Tables and reports are generated based on the input data to the system, ex. driving speed of drivers, passenger booking, passenger feedback etc. This sort of visualization of the data helps in easily understanding the situation as a whole, the trends etc.

9. Challenges and Open Issues:

1) Identifying and including different entities or communities to include them in communication with the management system. These entities include hospitals, health care centers, law enforcement authorities.

2) Filtering and analysis of data input to the system and sending correct messages to the appropriate entity (eg. Entities specific to a region - local law enforcement authorities, nearby hospital or health center).

3) Ensuring that all the cars in the fleet are tracked (some drivers may leave, cars may go out of service, new cars may get added to the system)

4) Storage and data processing capabilities within the data center should be high as there is lot of data in the system.

5) Data collected from sound sensors – accurate determining when the sound recorded could indicate a problem in the car (for example, an accident)

6) Data center is only place for the cloud computing. It is not distributed among various locations. Hence data center can prove to be a point of failure or a bottleneck. The system needs to be expanded to a distributed system with centers at multiple locations.

7) Every driver in the fleet must be trained to use the mobile app so that the driver can react to the notifications/alerts.

References:

1. IPv6 Primer
2. Crash Sensors http://www.safercar.gov/Vehicle+Shoppers/Air+Bags/Crash+Sensors
3. Different protocols used in IoT http://electronicdesign.com/iot/understanding-protocols-behind-internet-things
4. 6LoWPAN http://orbigo.net/2011/06/6lowpan-second-part-more-concepts-about-6lowpan/?ckattempt=1