FINAL ARCHITECTURE DESIGN DOCUMENT

Trafi Track

A Vehicle tracking and Taxation system

Architecture Design Document By

Mika Kivijoki
Syed Islam
Habibul Islam

Turun yliopisto
University of Turku

April 2017

Table of Contents

1. Introduction

The proposed name of the system is **Trafi Track**, which unify the vehicle tracking and tax management operation into centralized web-based service. The goal of the system is to give tax authorities the ability to collect taxes based on vehicle usage. System will also provide means to monitor vehicle owner based road usage and provide data for billing purposes. Moreover, the system serves vehicle tracking information and other location based information about vehicle usage for different kind of vehicle users. Vehicle tracking system also supports automatic emergency calling system standard eCall.

This document is intended to visualize and convey the significant architectural decisions in order to represent the overview of vehicle tracking system for small nordic country. By analysing the architectural overview, people involved in the project can better understand the problems to be solved and how it will be represented with this system. Likewise, the final application must be efficient and very easy to use. In this document, we also describe the physical architecture of the system that must be constructed by using GPS for positioning information and GSM/GPRS for transmission of the information. However, the system should provide following functionalities beyond general design principle:

- Determining the most effective connectivity scheme
- Service testing
- Support for different stakeholders and transports
- Users, bills and transport history
- Data validation
- Users privacy on confidential information
- Push notifications
- Automatic data synchronization
- An efficient Web user interface

1.1 Scope

The scope of this software design document (SDD) is to depict the architecture of the Vehicle Tracking System (Trafi Track) web application that will be implemented for a small nordic country. This document describes the aspects designing the target web application that are considered to be architecturally significant. This design document is intended for different stakeholders who will be working with this Trafi Track. After reading the document, stakeholders will be able to understand the system and also will become familiar about necessary elements and behaviors that are most fundamental for guiding the construction of this Trafi Track web application.

1.2 Assumptions

Because of the limitation of time and resources we aimed not to design a vehicle OBU unit, rather we assume that the car manufacturers will sells car in the target country who intend to use this system, must be equipped with OBU that gives access to read vehicle data such as car location data, mileage data, and ability to call in case of any emergency or accidents. The system implementation should be done by keeping the following assumptions in mind:

OBU registration, login and logout: Car manufacturers will be given a firmware software to be installed on each vehicle's OBU unit so that OBU units can be registered to the system. This way each new vehicle will be introduced to the system. The system can connect to the target system whenever a vehicle's start key was ignited. OBU unit will be disconnected from the system whenever the car was stopped. This way each vehicle or OBU unit will act as an actor and they will be login and logout of the system.

OBU data: An OBU unit feeds two types of data to the system: GPS location data, and vehicle mileage data. OBU units are also able to make emergency calls automatically whenever any accidents happen. Additionally a driver can also make an e-call by pressing a button in the car. The button is an interface provided by the OBU unit and it will make e-call and the call will be forwarded to the designated call center. Automatic e-calls will be triggered whenever the airbag comes out. Therefore, we assume that the car manufactures will provide a button for making an e-call as well as the OBU should be able to detect the collisions as like as the air-bags.

OBU components: To be able to make a call the OBU unit should also be equipped with a GSM modem that will allows the OBU to connect to the GSM/3G/4G and upcoming 5G networks. It is important to note that, making an e-call does not necessarily require to interact with the target system as the e-calls will be made directly to a call center where human actors will answer and help with the emergency situations. In order to be able to send location data the OBU should also contain a GPS tracker module inside of it. The GPS tracker will update and resend location data every 30 seconds.

OBU firmware: Any update for OBU firmware will be sent directly from the system. Assuming that admin will send the update file to the OBU. The OBU will then download the file and it will be updated with the new firmware whenever the vehicle is restarted. Old firmware files will be deleted consequently.

1.3 Definitions, Acronyms, and Abbreviations

Trafi Track- Vehicle Tracking System

OBU- On Board Unit

GSM- Global System for Mobile Communications

GPRS- General Packet Radio Service

DB- Database

DBMS- Database management system

DMZ- Demilitarized zone

IDS- Intrusion Detection System

HIDS- Host-based Intrusion Detection System

NIDS- Network-based Intrusion Detection System

SDD- Software Design Document

GUI- Graphical User Interface

Apache- Web server

Firmware- A software program that programmed on a hardware device
Firewall- Network security system monitors and controls the incoming and outgoing network traffic

1.4 References

The reference books and software design document materials that we followed to model all the architectural diagrams are:

[1] Alexander Kossiakoff, William N. Sweet (2011). *Systems Engineering: Principles and Practices*
[2] XML Legal Documents Utility Software Development Plan
[3] The "4+1" view model of software architecture, Philippe Kruchten, November 1995
[4] Documenting Software Architectures. Addison-Wesley 2003
[5] Server Load Balancing.Tony Bourke August 2001

2. Architectural Design Patterns

2.1 Architectural style

MVC (Model View Controller) design pattern has been mainly followed to describe the Trafi Track system. Technically, it breaks down the given application into Model, View and Controller in order to separate internal representations of information from the ways that information is presented to and accepted from the user. This is one of the most suitable architectures for designing web applications. In a nutshell, the main three parts are responsible for following tasks.

- **Model -** This is in charge of maintaining data.
- **View -** This part is in charge of displaying the data to the user.
- **Controller -** This is the Backend functionality or mainly the Software Code that controls the interactions between the Model and View

One of the factors which has led to choose this design pattern is its adaptability. That is to say, the design is closed for changes and open for additions. With this in mind, it is best suitable for applications in which there is a likelihood to add more business entities with new requirements in future.

2.2 Architectural model

There are several architecture models available that are used widely such as 4+1 model, C4 model and so on. In order to make the stakeholders to understand the target system better, we opted for a custom model which will use four different types of architectural views in order to describe our system in most understandable manner.

Stakeholders: – Architect – Web Developer- Web Designer- Administrator – Tester – Expert for standards – Security manager – Project manager – Network Architects- Database admin- Product manager – Customer – End user – Application area expert – Maintenance – Marketing – Program developer – Hardware expert – Ancillary service manager- System Designer- System Administrator- Software Developer- Network Engineer- Programmers

The four architectural views are:

- **Logical Architectural View- Context:** It mainly describes the relationships, dependencies, and interactions between the Trafi Track system and its environment. The overall environment basically describes the design object model including the people, systems, and other external entities. The topology describes the overall graphical representation of the "flow" of data through an information system to the end user.
  Audience: System Designer, Network Engineer, Application area expert
  Related Artifacts: Data flow model, Context Design model

- **Scenario View- Use- Case:** The Use-Case represents a discrete unit of interaction between a user (human or machine) and the system. It clarify the functionality that needs to be established in the proposed system. Moreover, it also include another Use Case's functionality or extend another Use Case with its own behavior.

Audience: All the stakeholders of the system, including the end-users.

Related Artifacts: Use case Model, Use-case documents

- **Development View- Component:** It is one of the major parts of our design model. However, the component view, does not itself describe the functionality of the system but mainly describes the components used to make those functionalities. With component view we illustrate the structure of this overall Trafi Track systems. Here it describes the organization and relationships of the components

  Audience: Programmers, System admin, Web Designer

  Related Artifacts: Implementation model, components

- **Physical View: Deployment:** In order to describe the environment into which the system will be deployed, including capturing the dependencies we used the deployment view to show the hardware environment that the system needs. It maps the technical requirements such as network interconnection, storage facilities, processing nodes, servers model and software model the system needs during runtime environment.

  Audience: System Engineer, Software Developer, System Admin, Database Admin, Network Architects, Security Specialist, Software Engineers

  Related Artifacts: Deployment model

3. Operational Goals and Constraints

3.1 Client Side

Users will be able to access Trafi Track application only through the internet. Clients/ or users need to use any web browser such as Mozilla Firefox 10, Internet Explorer 9, latest versions of Google Chrome or Safari to access the system. Users can also able to connect from their mobile phones. Therefore supports and consideration for using mobile browsers also taken into account.

3.2 Server side

The target system will be hosted on a Web application cluster server that are running on Apache 2 web server. A cluster is a group of servers running a Web application simultaneously, appearing to the world as if it were a single server, thus maintaining the cloud server attitude. More application servers and database servers can be added on the fly. All communication with client has to comply with public HTTPS, TCP/IP communication protocol standards. The hosting server will consists of the following hardwares:

- **Separate Database Server with Master-Slave Database Replication:** We aimed to separate the DBMS from the application server to eliminate resource contention between the application and the database. It will also allow us to increase security by hiding the database from DMZ (sometimes referred to as a perimeter network), or public internet. The Trafi Track will be a such system where more read operations than writing will be requested. In order to improve the DBMS read operation performance we will use use master-slave database replication. This will allow the database system to perform many reads rather than writes. At the same time all updates are sent to the master node and allowing the read requests to be distributed across all nodes. In Trafi Track web server the Master-slave replication system requires to install one master and one or more slave nodes depending on the resources.

- **Load Balancer (Reverse Proxy):** Load balancers can be added to a server environment to improve performance and reliability by distributing the workload across multiple servers. If one of the servers is unable to handle requests or is down for some reason, the other servers will handle the incoming traffic until the failed server becomes live again. Available tools that can be used to setup load balancing server are HAProxy, Nginx, Varnish, and KEMP. However we recommend to use Kemp which can handle 1M requests at a time. Some of the advantages that a load balancer can provide are horizontal scaling and protection from DDOS attacks. Horizontal scaling can be gained by adding multiple servers to improve environment capacity. Protection from DDOS attacks can be achieved by limiting client connections to a sensible amount and frequency.

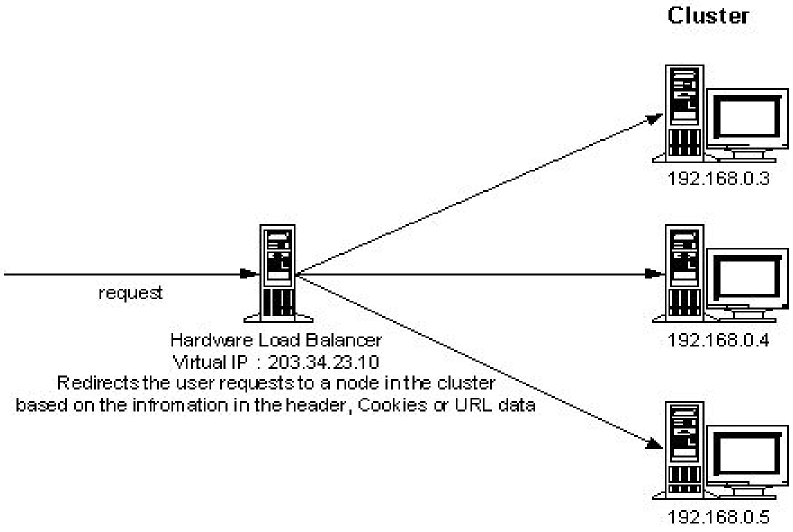
Cluster
192.168.0.3
request
192.168.0.4
Hardware Load Balancer
Virtual IP : 203.34.23.10
Redirects the user requests to a node in the cluster
based on the infromation in the header, Cookies or URL data
192.168.0.5

Figure 1: Hardware Load Balancer (Source Oreilly)

- **Hardware SSL decoders:** HTTPS requests are encrypted. This is why it's difficult to load balance and maintain session information of requests that come in over HTTPS. The hardware load balancer cannot redirect requests based on the information in the header, cookies, or URL readings. Two options available to mitigate this problem: Web server proxies, and Hardware SSL decoder. Because of the reason that hardware SSL decoders are faster and more efficient than Web server proxies, we considered to use Hardware SSL Decoder instead. It is important to note that, Hardware SSL decoders are costly and require considerable amount of efforts to configure. The figure below shows a typical installation of a Hardware SSL Decoder with a cluster.

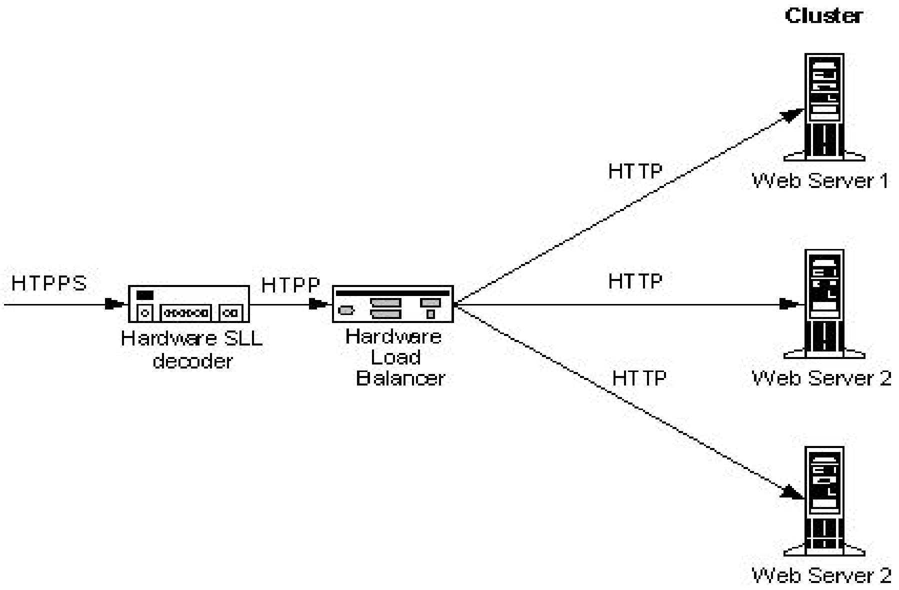

Figure 2: Hardware SSL Decoder setup with cluster (Source Oreilly)

- **HTTP Accelerator (Caching Reverse Proxy):** HTTP accelerators which is also known as caching HTTP reverse proxy allows the web server to reduce the time it takes to serve content to a user. Because the Trafi Track system will have files that are very common to same users. Therefore, employing an HTTP accelerator will enable caching responses from a web application server in memory. This way number of interaction with the web servers can be reduced. At the same time, any future requests of the same content can be provided time efficiently. Following are the examples of tools that can be taken into use to enable HTTP acceleration: Varnish, Squid, Nginx.

- **Intrusion Detection Systems:** An Intrusion detection systems or IDS detects attacks by sniffing at network traffics or operating system events. There are two types of IDS available: Network-based Intrusion Detection and Host-based Intrusion detection. Using a single approach for intrusion detection is insufficient as NIDS . Therefore, we will employ both types of IDS in the system. In a network-based IDS one network node screens HTTP traffic before it reaches the destination. An

NIDS works on the TCP/IP level and is used to detect attacks against any network service, including the web server. The target system will place an NIDS in front of the SSL decoder so that it can detect any malicious request before the request reached to SSL decoder. On the other hand, in a host or web server-based IDS, an intrusion detection agent is embedded within the web server. HIDSs typically work on the host level. Host-based intrusion is mostly concerned with the events that take place on the host (such as users logging in and out and executing commands) and the system error messages that are generated. HIDS can be implemented within a single script that will keep an eye on the log files for error messages. We recommend to use Integrity validation programs such as Tripwire as part of the HIDS. The target system will have an embedded HIDS in every application server.

- **Intrusion Prevention system:** The term intrusion prevention is used to refer to systems that are also capable of preventing attacks. Trafi Track system will have a Intrusion Prevention controller component which can be accessed only by the system Admin. Therefore, it is a component in Admin controller which is known as Intrusion Management that allows the admin to see the logs files sent by the HIDS and then restrict, prevent or mitigate those attacks through the IPS.

The example below shows a complete setup of the target web server.

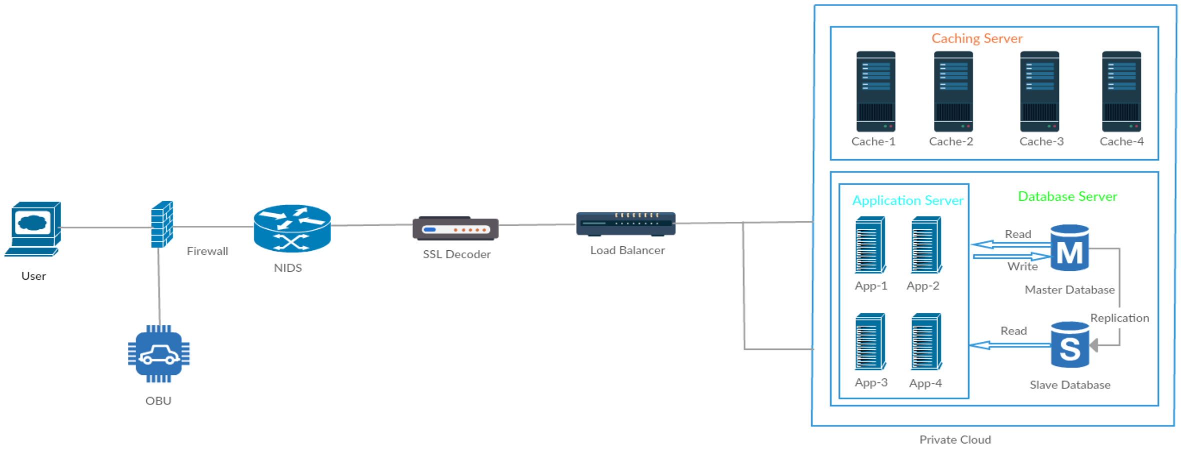

Figure 3: Example Trafi Track web server

3.3 Security

System security is mostly defined with the terms: **confidentiality, integrity** and **availability**. To achieve **confidentiality** and data **integrity** we suggest to use cryptography while exchanging data between users and the web server. Various concepts and techniques must be combined to achieve the full confidentiality. There are four important encryption techniques that should be taken into account:

- Symmetric encryption. Example symmetric encryption algorithms includes Data Encryption Standard (DES), Triple-DES (3DES), Blowfish, International Data Encryption Algorithm (IDEA), RC4, Advanced Encryption Standard (AES)
- Asymmetric encryption: Example asymmetric encryption algorithms includes Rivest, Shamir, and Adleman (RSA), Digital Signature Algorithm (DSA), Elliptic curve
- One-way encryption: Message Digest algorithm 5 (MD5), Secure Hash Algorithm 1 (SHA-1), SHA-256, SHA-384, and SHA-512
- Digital certificates: an electronic document used to identify an organization, an individual, or a computer system. A digital certificate must be generated for the web server.

The web server must setup and configure Secure Sockets Layer (SSL) which is a hybrid protocol that uses many of the cryptographic techniques mentioned earlier to establish secure communication layer. The official name of the standard is Transport Layer Security (TLS). Support for SSL is included with the Apache 2 distribution in order to secure HTTP. Successfully configured SSL and CA certificates should give the system protection from Man-In-the-Middle attack.

3.4 Persistence

Persistence is “the continuance of an effect after its cause is removed”. In the context of storing data in a computer system, this means that the data survives after the process with which it was created has ended. In other words, for a data store to be considered persistent, it must write to nonvolatile storage. The target system will take Disk-based data storage with update-in-place writes, such as MySQL.

3.5 Scalability / High Availability

Scalability and high availability is one of the most important concern for any enterprise Web application. Scalability is an application's ability to support a growing number of users. Scalability is a measure of a range of factors, including the number of simultaneous users a cluster can support and the time it takes to process a request. High availability can be defined as redundancy. In a highly available system, if a single Web server fails, then another server takes over, as transparently as possible, to process the request.

As mentioned earlier, in order to balance the server load and to optimize system performance, Trafi Track web server will employ a Hardware SSL Decoder and a Hardware Load Balancer that sits right at the front of the application servers. The load balancer distributes requests to different nodes within the server cluster which will results in higher availability and better scalability. Usually a web server takes 10 milliseconds to respond to a request. Let’s say Trafi Track will also take 10 milliseconds to respond to a login request. If the number of login requests exceeds more than 100,000 then the system should still be able to respond to the requests within 10 milliseconds with negligible degradation of performance.

3.6 OBU Tampering

In order to protect the OBU from being tampering physically, the OBU cover should be sealed off by the Vehicle Inspector. The seal has to be change every year when the vehicle visits the inspection center for its fitness inspection. Vehicle inspectors will also check mileage data in both OBU unit (accessible from the system) and in car dashboard meter. If the two readings are not matched than it should be assumed that the vehicle’s mileage data was tampered and the car should be banned from the system .

3.7 User's Privacy and Security

Data confidentiality is important for users to store their private or confidential data in the cloud. The user’s privacy and security challenges in the Trafi Track system includes threats, data loss, service disruption, outside malicious attacks, and multi-tenancy issues. The data confidentiality, authentication, and access control issues in the system could be addressed by implementing the private cloud system with reliability and trustworthiness. Authentication and access control strategies are used to ensure data confidentiality. Encryption is used to ensure the confidentiality of data at each stage of the communication. Database encryption technique also proposed for the privacy and security of the user's data in the environment. However, user security is handled by authenticating all users of the system and providing data only those user groups that has authenticated and approved access to data. All things considered, banking authentication was proposed in order to verify the user's identity while accessing the system.

3.8 Performance

A successful implementation of the target system will ensure to provide following performances:

- Eliminate resource contention between the application and the database
- Fast database read operations
- Better scalability with Horizontal scaling
- High availability
- Data Integrity
- DDOS protection
- Increased throughput with caching content
- Man-in-the-middle attack protection
- Intrusion detection and prevention

4. Logical Architectural Representation

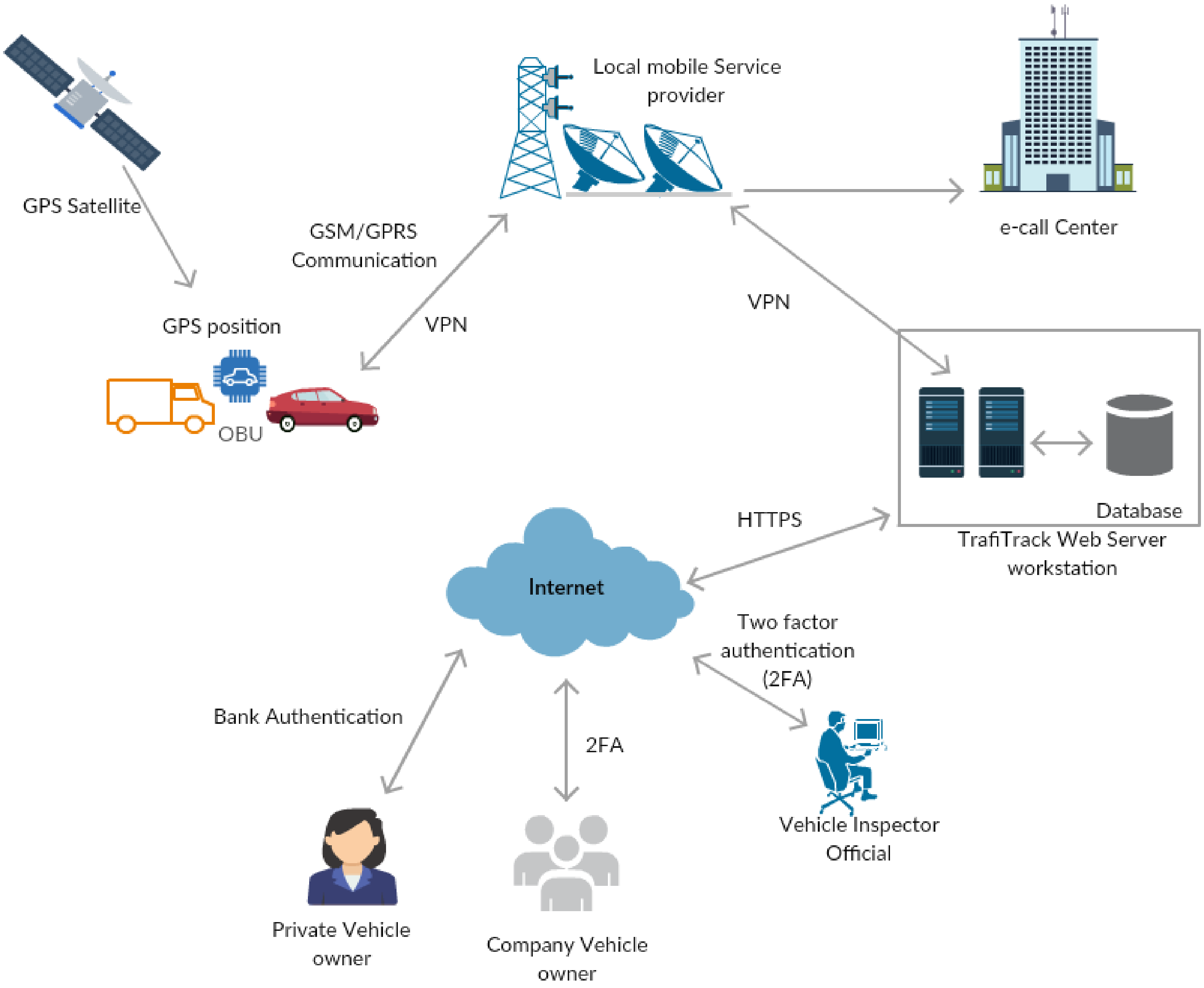

Figure 4: Context diagram of the Trafi Track system

collecting the location data along with other predefined parameters from the OBU, it is then transmitted to the remote base station or remote data center using most viable service GPRS on GSM network in real time. The vehicle’s location is determined using GPS and since it is a commonly used technology, we assume that transmission mechanism is most effective and economical through satellite, terrestrial radio or cellular connection from the installed OBU in the vehicle to nearby cellular network tower. The whole system is mainly integrated with web-based application, which is interfaced with Trafi Track GUI in order to provide applicable service to the different end users. The information is transmitted using TCP/IP connection with the the server through GPRS. The server should have dedicated a secured socket in this communication. The GPRS network includes the fixed network elements and their physical connections that convey OBU data and signalling information. Another key thing to in the system is emergency call, which is handled over the GSM network to the e-call service center.

The legitimate information is available only to the authorized users of the Trafi Track system through secured internet connection. The web based application mainly accommodate information such as vehicle status, owner's private data, milage driven, transportation history, billable tax amount and so on and this is core of this vehicle tracking system.

5. Scenario View: Use-Case

5.1 Actors

The target system primarily has six actors who will be interacting with the system. The actors are listed as below:

- Admins: Admins are actor with an access level of 1 that entails them to add and remove vehicle inspectors and also updating the vehicle OBU’s.
- Vehicle Inspector: Vehicle inspectors are actors who have access level of 2. Vehicle inspector can register a vehicle, update vehicle fitness status and also can take a car out of the road permanently (dumping car).
- Tax authorities: Tax authorities is an actor with an access level of 3. Tax authorities is only capable to search vehicles and check how much the vehicle has driven for a one whole year so that they can max road tax invoices for the respective vehicles.

- Private car owners: Private car owners are the most common actors of this system. A private car owner has an access level of 4 which entails them to ordering vehicle registration and taxation information, real time vehicle tracking, take off the vehicle from road and put the vehicle back to the road etc.
- Company car owners: A company car owner is another actor who has same access level and functionalities as the private car owners.
- Vehicle OBU: The OBU of each vehicle represents the vehicle as an actor to the system. OBU's have access level of 6 and these are actors that interact with the system more than any other actors. Each OBU send location and mileage data to the system and receives firmware update from the system.

5.2 Use cases

We have identified fifteen use cases for the Trafi Track. The use cases are listed as follows:

- Order vehicle registration certificate
- Order vehicle taxation
- Vehicle commissioning
- Vehicle decommissioning
- Track vehicle location
- View vehicle mileage
- Search vehicle
- Add vehicle Inspector
- Remove vehicle inspector
- Update vehicle OBU
- Send location data
- Send mileage data
- Update vehicle fitness status

- Register vehicle
- Dump vehicle

Interaction between the actors and system through the use cases are depicted in the following use case diagram.

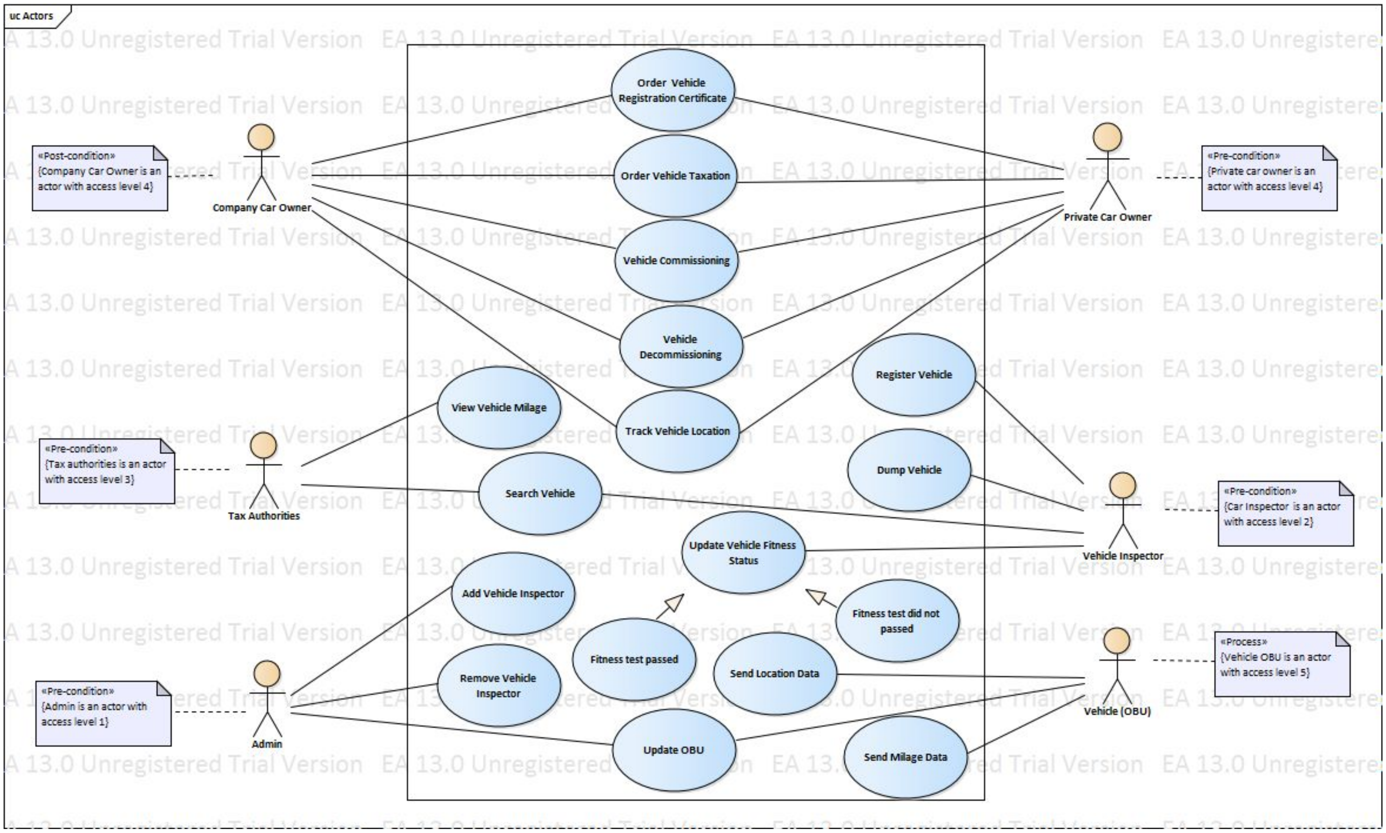

Figure 5: Use Case Diagram

Detailed view of the Use-Case diagram available at this link:

https://drive.google.com/open?id=0B8_j8N9MO33pZHd6WkhWT1NMTWs

6. Development View: Component

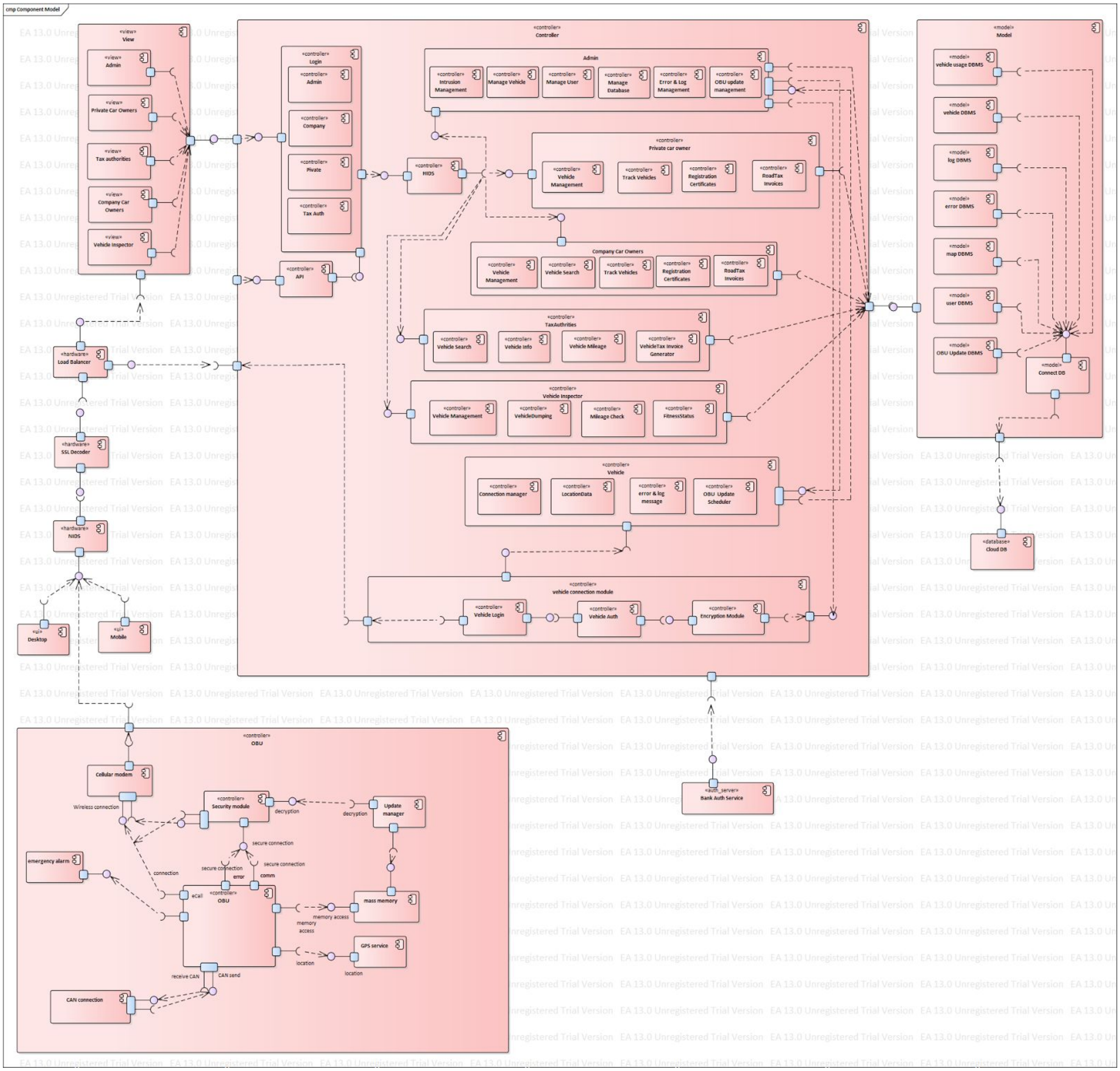

Figure 6: Component diagram

Detailed view of the component diagram available at this link:

https://drive.google.com/open?id=0BxX2QLFssAxGTWZ0cEZJVjdCeWM

6.1 Control part of the software components

- **Login** provides login authentication for all the user groups
- **HIDS** provides host based intrusion detection
- **Admin** components provides practically all functionality such as Managing the system, Intrusion management system, vehicle Inspector registration, deleting/ updating the system
- **Private car owner** consists components required to track own vehicle and view stored location and vehicle data
- **Company car owner** consists components required to track multiple company vehicles and view stored location and vehicle data
- **Tax Authorities** group consist components are need to run individual queries and automated vehicle usage queries to whole vehicle base. They are also entitled to generate the bills
- **Vehicle inspector** component group provides functionality to vehicle inspector to add and remove vehicle. Vehicle inspector can also add vehicle fitness status
- **Vehicle component** group consists component that handles connection to vehicles, handles update scheduling and error and message logging
- **Vehicle connection** component provides authentication, login and encryption to vehicle connection
- **VIEW components:** provides Web user interfaces for all user groups
- **Vehicle Dumping** only applicable when the vehicle is not usable anymore
- **Bank Authentication Service** provides interface to 3rd party authentication method called Tupas, which is used to provide authentication for private users as well as the primary authentication for company car users for two factor authentication
- **Cloud DBMS** stores all data and deliver them to users on demand via the Internet from a cloud database provider's servers
- **Two factor authentication** provides authentication to different users.

  -Company users must be enrolled with specific mobile numbers and ID

  - Every time they login it requires a pass code that is sent to authenticated mobile number

  - With valid pass code company’s user can access to the system

- **OBU Software Components**

  - GPS service provides vehicle location data to the system

  - Cellular modem provides common interface to wireless transceiver

  - Emergency alarm component provides automatic and manual trigger, which initiates eCall alarm

  - security module provides encryption and decryption services for other functions

  - mass memory provides enough memory to handle software updates and temporary memory space for logging and location data

  - update manager handles the OBU update process with the server

  - error & log creation forms messages

  - optional CAN connection component provides interface for CAN bus, if implemented

  - OBU component handles almost all functional control in onboard unit
- **Trafi Track Workstation** is the unit which consists of all the servers and database system to handle smooth running of the application
- **Web Server mainly** host the Trafi Track web application and supply web queries

6.2 OBU software update

Components handling the OBU software update are shown already in component diagram of whole system, but they are also reproduced on following diagrams to give clearer view of components concerning OBU updates. New software update packages are placed to database by using OBU update management component from administrator component group. Actual software update details are also added by using same OBU update management component. OBU update management component sets the update parameters for OBU update scheduler which handles the actual update process with the OBUs in the background. Update scheduler downloads update package from update database and contacts the required OBUs and forms a connection with OBU’s with help of vehicle connection module. Update scheduler informs OBU units that new update package is available.

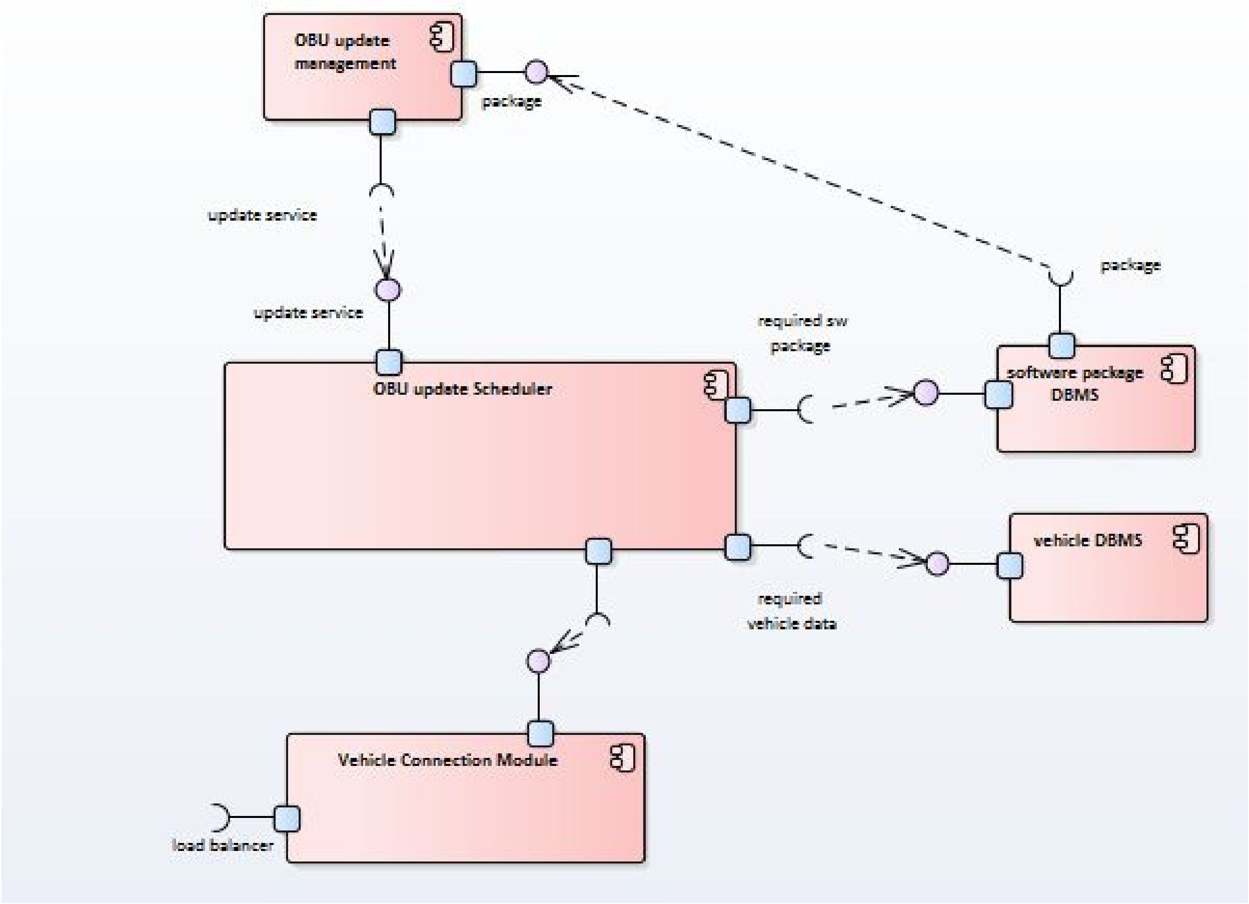

Figure 7: Server components related to OBU software update

After the secure connection is established, individual OBU starts to download an update package from update scheduler. Download is handled by update manager in OBU, Update manager first downloads update package to mass memory and checks the update package validity. If package is valid, update manager uninstall previous software version except bootloader section. New program is installed from mass memory. If program is successfully installed, then the old installation package version can be removed.

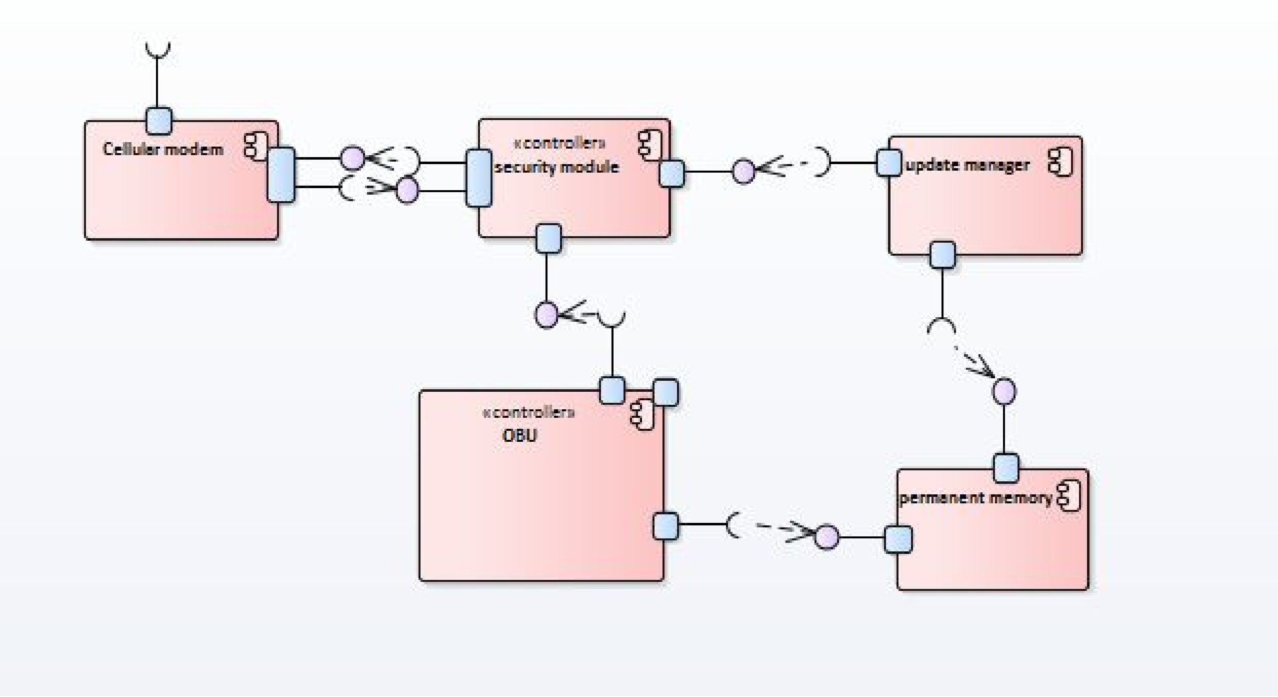

Figure 8: OBU components related to software update

6.3 Logging and error messages

When OBU controller detects malfunction or error on one of its subcomponents it creates error message. OBU then forms a secure connection to a server and then sends the error message to server.

When server receives message from OBU unit it will be at first handled by error & log message component, which is in the vehicle component. Error & log message component will determine if message is determined as an error and will rise error in error & log management component in Admin component group and will adds also error message to error database. If message is determined as normal logging message, it is then saved to log database.

7. Physical View: Deployment

Vehicle tracking system allows access for several user types to the system. Each user type has a different user interface, but user interface comes from same server. Correct user interface is provided to user after the user has logged into the system and user credentials are confirmed.

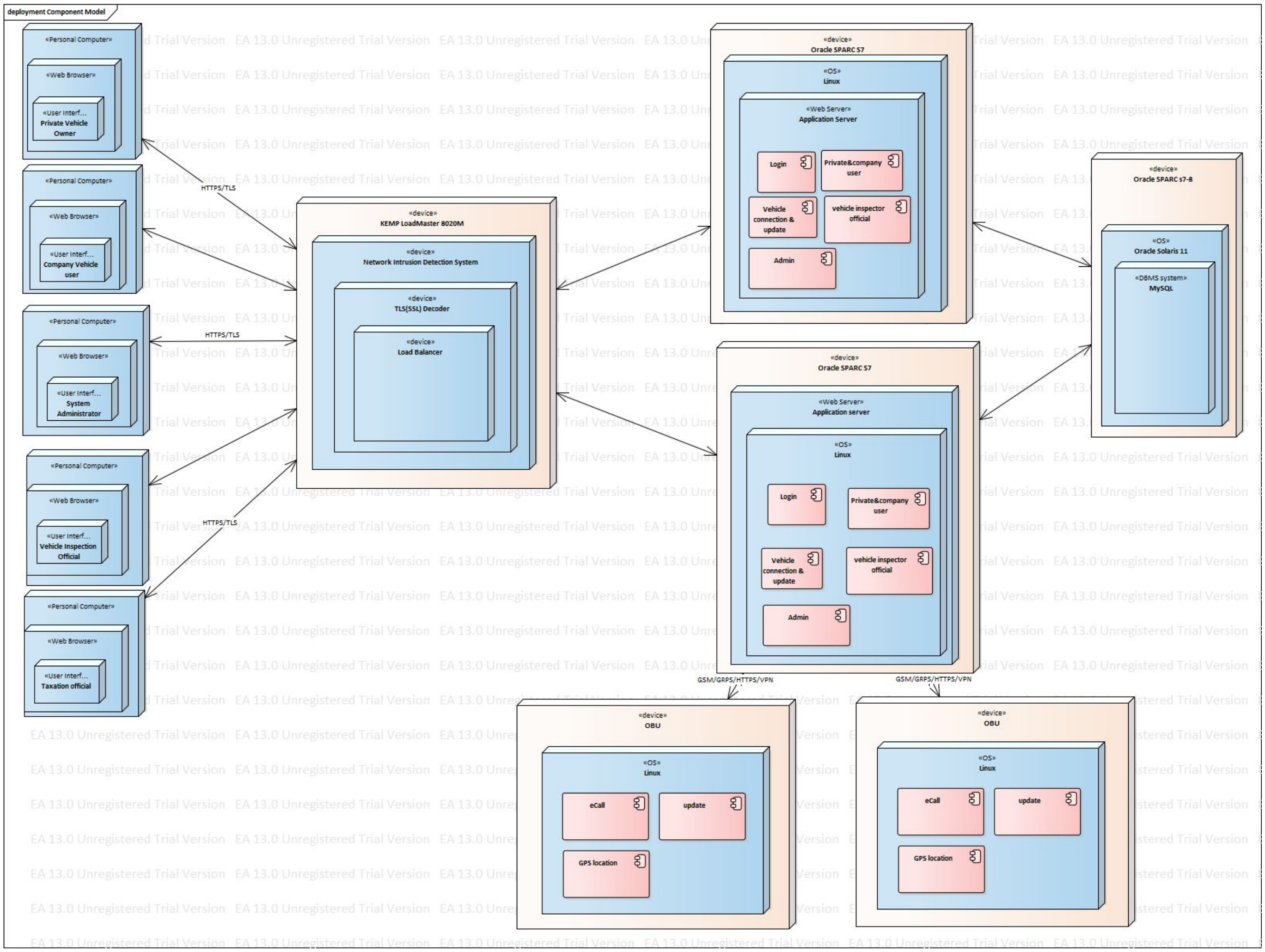

Figure 9: Deployment diagram

Detailed view of the deployment diagram available at this link:

https://drive.google.com/open?id=0BxX2QLFssAxGT3EtR3c3ZGN3UDQ

Network Intrusion Detection, SSL decoder and Load Balancing are all implemented with one hardware accelerated unit. SSL decoder is providing practically all control logic related to user interfaces is handled by the web server. Web server handles connections to OBU's and databases. Web server relays location data from OBU's to databases and controls OBU unit software updates. Web server also collects logging and error data from OBU's and relays that data to database. Database deployment unit contains databases for user data, location data, software updates, vehicles, etc.OBU or On Board Unit handles recording of GPS location data and sends that data to web server. OBU has ability to make also automatic or manual emergency call according to European eCall specification. OBU is able to also handle over the air (OTA) software updates.

8. Issues and concerns

In order to make the system design more understandable to the respective stakeholders further revision of this architecture design document can be enhanced by adding class diagrams for each components. We tried to make the system as much secure as possible. However achieving 100% security is never possible in real world. Therefore, we also recommend to construct data flow diagrams for every data entity so that more security measures can be taken while different components exchanges different types of data among them. Development team is advised to perform a threat modeling approach followed by a threat library that includes most common vulnerabilities. Additional and system specific vulnerabilities can added to the threat library after analyzing the data flow diagrams.

9. Future Work

- API for more actors such as Police
- Road navigation system can be implemented so that cars can know which road is busy
- Human factors engineering outputs - what kind of possible error can happen during the smooth running of the system.

- There is an optional software component provided to connect OBU unit to vehicle CAN bus. This connection can be used to connect vehicle tracking system to vehicle’s information system and provide new uses for the data provided by the system.