BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT

PROJECT REPORT

Submitted by

AROCKIA SNEHA A	**(211417104022)**
HARINI S	**(211417104079)**
KAVIYA B	**(211417104116)**

in partial fulfillment for the award of

degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600 123

ANNA UNIVERSITY, CHENNAI- 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT**" is the bonafide work of "**AROCKIA SNEHA A (211417104022), HARINI S (211417104079), KAVIYA B (21141710116)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**Mrs.C.VIJAYALAKSHMI,M.Tech.,Ph. D**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI- 600 123	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI- 600 123

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on **05.08.2021**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,Ph.D,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mrs.C.VIJAYALAKSHMI, M.Tech., Ph. D.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

AROCKIA SNEHA A
HARINI S
KAVIYA B

ABSTRACT

The Internet of Vehicles (IoV) provides new opportunities for the coordination of vehicles for enhancing safety and transportation performance. Vehicles can be coordinated for avoiding collisions by communicating their positions when near to each other, in which the information flow is indexed by their geographical positions or the ones in road maps. Vehicles can also be coordinated to ameliorate traffic jams by sharing their locations and destinations. Vehicles can apply optimization algorithms to reduce the overuse of certain streets without excessively enlarging the paths. In this way, traveling time can be reduced. However, IoV also brings security challenges, such as keeping safe from virtual hijacking. In particular, vehicles should detect and isolate the hijacked vehicles ignoring their communications. The current work presents a technique for enhancing security by applying certain prioritization rules, using digital certificates, and applying trust and reputation policies for detecting hijacked vehicles. We tested the proposed approach with a novel agentbased simulator about security in Internet of Things (IoT) for vehicle-to-vehicle communications. The experiments focused on the scenario of avoidance of collisions with hijacked vehicles misinforming other vehicles. The results showed that the current approach increased the average speed of vehicles with a 64.2% when these are giving way to other vehicles in a crossing by means of IoT.

LIST OF FIGURES

LIST OF ABBREVATIONS

JDK	Java Development Toolkit
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool
VANET	Vehicular Ad-hoc Network

CHAPTER 1

INTRODUCTION

Aim:

The main aim of this project is providing the secure communication between the two cars in VANET environment.

Overview:

The Internet of vehicles (IoV) provides new opportunities for the coordination of vehicles for enhancing safety and transportation performance. Vehicles can be coordinated for avoiding collisions by communicating their positions when near to each other, in which the information flow is indexed by their geographical positions or the ones in road maps. Vehicles can also be coordinated to ameliorate traffic jams by sharing their locations and destinations. Vehicles can apply optimization algorithms to reduce the overuse of certain streets without excessively enlarging the paths. In this way, traveling time can be reduced. However, IoV also brings security challenges, such as keeping safe from virtual hijacking. In particular, vehicles should detect and isolate the hijacked vehicles ignoring their communications. The current work presents a technique for enhancing security by applying certain prioritization rules, using digital certificates, and applying trust and reputation policies for detecting hijacked vehicles. We tested the proposed approach with a novel agent-based simulator about security in IoT for vehicle-to-vehicle (V2V) communications (ABS-

SecIoTV2V). The experiments focused on the scenario of avoidance of collisions with hijacked vehicles misinforming other vehicles. The results showed that the current approach increased the average speed of vehicles with a 64.2% when these are giving way to other vehicles in a crossing by means of IoT.

Problem Definition:

When vehicular networks are open to other heterogenous communication, these usually raise many security challenges.

Some of these challenges are related with the location accuracy, location verification, location privacy, and the operational management of all the traffic of the different networks.

CHAPTER-2

LITERATURE SURVEY

[1]

TITLE: A Novel Clustering Algorithm Based on Agent Technology for VANET

AUTHOR: Samira Harrabi,
Ines Ben Jaafar

OBJECTIVE:

Goal of VANET is to provide communications between nearby nodes or between nodes and fixed infrastructure. Various searches have been recently published deal with clustering for VANETs. But most of them are focused on minimizing network overhead value

DISADVANTAGE:

The most proposed clustering algorithms for MANET are unsuitable for VANET. Number of created clusters and had not considered the vehicles interests which defined as any related data used to differentiate vehicle from another.

[2]

TITLE: Agent- Based Tools for Modeling and Simulation of self Organization in Peer-to-Peer, Ad Hoc and other Complex Networks

AUTHOR: Muaz Niazi
Amir Hussain

OBJECTIVE:

Agent-based modeling and simulation tools provide a mature platform for development of complex simulations.Simulation of complex networks such as pervasive computing, large-scale peer-to-peer systems, and networks involving considerable environment

DISADVANTAGE:

They however, have not been applied much in the domain of mainstream modeling and simulation of computer networks.To point out problems with the current simulation methodologies for telecommunications networks, such as the use of pseudo-random number generators.

[3]

TITLE: Routing Protocols in Vehicular Ad hoc Networks: Survey and Research Challenges

AUTHOR: Kayhan Zrar Ghafoor
Marwan Aziz Mohammed

OBJECTIVE:

A Vehicular Ad hoc Network (VANET) is a type of wireless ad hoc network that facilitates ubiquitous connectivity between vehicles in the absence of fixed infrastructure. We analyze the simulation results and discuss the strengths and weaknesses of these routing protocols in regard to their suitability to vehicular networks

DISADVANTAGE:

Multi-hop routing and beaconing approaches are two important research challenges in high mobility vehicular networks. We perform a comparative study among the existing routing solutions, which explores the main advantages and drawbacks behind their design.

[4]

TITLE: Privacy-Enabled Probabilistic Verification in Broadcast Authentication for Vehicular Networks.

AUTHOR: kanika grover
Alvin lim

OBJECTIVE:

Vehicular Ad hoc Networks (VANETs) authentication schemes need to consider mobility and rapidly changing topologies in addition to an unreliable wireless channel communication.We propose a practical and efficient strategy that makes use of secure ECDSA but still decreases the computation time for the most relevant packets.

DISADVANTAGE:

It has the drawback of expensive computations for verification.We will implement and study IEEE 1609:2 on realistic VANETs and study the problem of long verification delay.

[5]

TITLE: Improving Relay Selection Scheme for Connecting VANET to Internet over IEEE 802.11p

AUTHOR: Driss Abada
Abdellah Massaq

OBJECTIVE:

Vehicular ad hoc networks (VANETs) enable vehicles to communicate with each other (V2V) as well as with roadside infrastructure units (V2I). Based on mobility parameters, we can select longest life time routes to routing packet, but these routes may be with low quality of signal.

DISADVANTAGE:

A high vehicular speed can cause frequent disconnection between nodes, the large number of vehicles on the road can flood the VANETs network by broadcasting messages which will increase overhead.

Congested and more realistic environment scenario, when the number of vehicular sources increases, interference, noise and access channel contention increase as well.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Vehicles with IoT and autonomous decisions on motion imply many challenges for the viewpoint of security and safety, as one can observe in the variety of possible attacks over self-driving vehicles [12]. If a vehicle is able to brake, turn or accelerate for avoiding a collision based on the information received by Internet, the car must completely validate the veracity of this information. Otherwise, a hijacked vehicle could provoke collisions or make other vehicles to unnecessarily stop. The hijacked vehicle would achieve this by intentionally sending wrong information to the other vehicles.

3.2 PROPOSED SYSTEM

We are proposing a emergency message in VANET environment. In this system messages are broadcasted only in their region of interest so that delivery latency will be decreased. In Proposed system if a vehicle in the environment detects a dangerous event, it immediately generates and broadcasts emergency message to vehicles in region of interest, so that the vehicles can take preventive measures to avoid accident. Emergency Messages will be broadcast to vehicles which are needed to take action to avoid accident. And we proposed the security function for this project while message communication. When the messages send to one user \to another user we should encode the content inside the message and compare the encoded values. If the encoded values are same that user recognized the authorized user and they will get the trust vales, otherwise that user recognized as an unauthorized user they will get a minus trust values.

3.3 REQUIREMENT SPECIFICATIONS

3.3.1 INTRODUCTION

SOME smart vehicles can interact among each other, by means of Internet of Things (IoT), conforming a new field named as Social Internet of Vehicles (SIoV) [1]. Manuscript received July 10, 2018; revised August 20, 2018; accepted September 16, 2018. Date of publication September 19, 2018; date of current version July 31, 2019. This work was supported by Harvard University (stay funded by T49_17R), University of Zaragoza (JIUZ- 2017-TEC-03), Foundation Bancaria Ibercaja, Foundation CAI (IT1/18), University Foundation Antonio Gargallo (call 2017), and "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" (TIN2017-84802-C2-1-P). *(Corresponding author: Jaime Lloret.)* I. García-Magariño and R. Lacuesta are with the Department of Computer Science and Engineering of Systems, University of Zaragoza (Teruel Campus), 44003 Teruel, Spain, and also with the Technological Systems in Health Field, Instituto de Investigación Sanitaria Aragón, 50009 Zaragoza, Spain (e-mail: ivangmg@unizar.es; lacuesta@unizar.es). S. Sendra is with the Departamento de Teoría de la Señal, Telemática Comunicaciones, Universidad de Granada, 18071 Granada, Spain (e-mail: ssendra@ugr.es). J. Lloret is with the Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politècnica de València, 46022 Valencia, Spain (e-mail: jlloret@dcom.upv.es) Vehicles can cooperate for conforming vehicular ad hoc networks (VANETs) with different routing protocols [2]. In addition, Internet of public transport vehicles can support VANETs by means of communication among vehicle groups that dynamically change [3]. In this context, vehicles can cooperate among each other 1) to avoid collisions; 2) to estimate the routes with least traffic; or 3) to arrange the best routes for avoiding waiting times in the charging stations for electric vehicles [4]. These are some of the most important issues that require the vehicles' data collection. These data should be transferred in a secure way with mechanisms like the existing one for big data collection from vehicles via a mutual authentication and single sign-on algorithm [5]. The security in the Internet of Vehicles (IoV) can also be useful for safely making the emergency rescue operations more efficient, and gathering reliable proofs of accidents, such as the speeds and positions of vehicles [6]. Vehicles can connect among each other through vehicle-tovehicle (V2V) communications and with the city infrastructure by means of vehicle-to-infrastructure (V2I) communications. Both kinds of communication can support real-time operations [7]. For instance, V2V communications can be useful for detecting traffic congestion in large-scale scenarios [8] or cooperating for car parking [9]. For example, V2I communication can support the stabilization of vehicle strings for reducing disturbances, with adaptive driving strategies [10]. In addition, vehicles can also use V2I communications in a street-aware and intelligent beaconless forwarding protocol for achieving fast and reliable communications in urban vehicular scenarios [11]. Vehicles with IoT and autonomous decisions on motion imply many challenges for the viewpoint of security and safety, as one can observe in the variety of possible attacks over self-driving vehicles [12]. If a vehicle is able to brake, turn or accelerate for avoiding a collision based on the information received by Internet, the car must completely validate the veracity of this information. Otherwise, a hijacked vehicle could provoke collisions or make other vehicles to unnecessarily stop. The hijacked vehicle would achieve this by intentionally sending wrong information to the other vehicles. There are several mechanisms for performing traffic simulations. For example, model-driven development can be used for this kind of simulation as in the work [13], which defines a domain-specific modeling language for defining traffic simulations. In addition, agent-based simulators (ABSs) have proven to be useful for simulating IoV. In this context, vehicles are modeled as agents, and V2V communications are simulated as social interactions among agents [9]. In this line, agent technology was proposed to improve the routing in

VANETs with a novel clustering algorithm [14]. Several works proposed different solutions for supporting authentication in vehicular networks. For instance, VANETs used authentication by means of the elliptic curve digital signature algorithm (DSA) in broadcast messages, considering privacy, probabilistic, and defense from DDoS attacks [15]. In addition, Scheme for IEEE 802.11p was improved for V2I communications with a lightweight authentication that focused on security and privacy of vehicles [16]. The trust management has proven to be useful in VANETs for maintaining security in VANETs considering probabilistic approaches, deterministic ones and combination of these [17]. Vehicles need to verify that the locations reported by other vehicles are both accurate and reliable. For example, Kasana *et al.* [18] improved the accuracy of locations by considering neighbor locations. In addition, Khalid *et al.* [19] used transferable belief models for ensuring security in the sharing of vehicle locations. When vehicular networks are open to other heterogeneous communications, these usually raise many security challenges, as reviewed by Kaiwartya *et al.* [20] when presenting their architecture for IoV. Some of these challenges are related with the location accuracy, location verification, location privacy, and the operational management of all the traffic of the different networks.

3.3.2 HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE REQUIREMENTS:

• Hard Disk	:	500GB and Above
• RAM	:	4GB and Above
• Processor	:	I3 and Above
• Kit	:	Arduino with Accelerometer Sensor

SOFTWARE REQUIREMENTS

- JDK 1.8
- Net beans 8.1

- Windows 7

TECHNOLOGIES USED

- JAVA
- JAVAFX

3.3.3 FUNCTIONAL REQUIREMENTS

System Features

In this context, the current work proposes an approach aimed at maintaining security and safety in vehicles with IoT. It combines proper authentication by asymmetric encryption, prioritization rules and management of trust and reputation over vehicles identifiers. The current approach is illustrated with a novel ABS about security in IoT with V2V communications (ABS-SecIoTV2V). The current article is organized as follows. The next section introduces the most relevant related work highlighting the gap covered by the current work. Section III presents the technique for achieving security in the IoV from virtual hijacking focusing on respectively the prioritization rules, the vehicle certificates and the trust management. It also presents the novel ABS about collision avoidance in crossroads for illustrating the current approach. Section IV presents the experiments that we conducted for assessing the current approach. Finally, section V mentions the conclusions and depicts some future research lines.

Design and Implementation Constraints

Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

Other Nonfunctional Requirements

Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;

- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

Safety Requirements

- The software may be safety-critical. If so, there are issues associated with its integrity level
- The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
- If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
- Systems with different requirements for safety levels must be separated.
- Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

CHAPTER 4

SYSTEM DESIGN

4.1 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

4.1.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

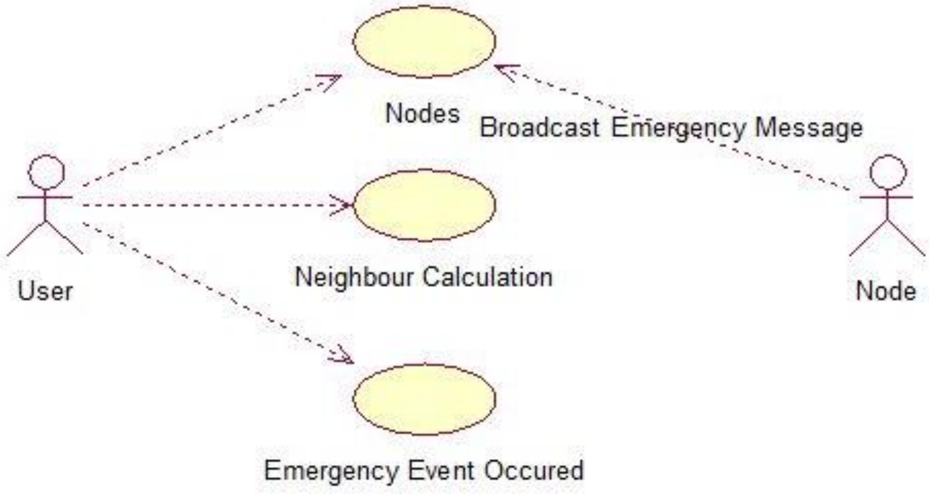


Figure 4.1: Usecase diagram

4.2 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

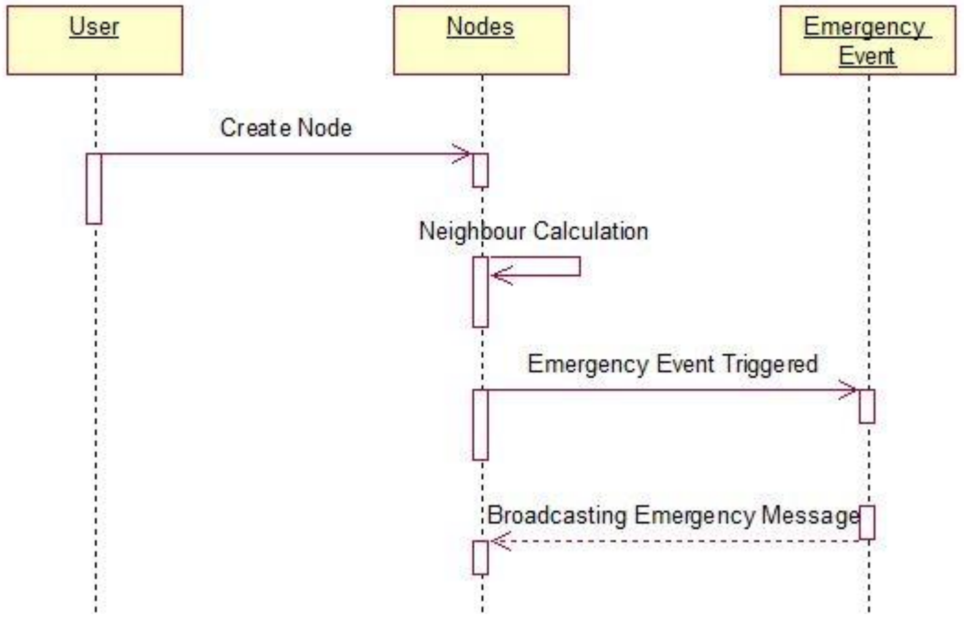


Figure 4.2 : Sequence diagram

4.3 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

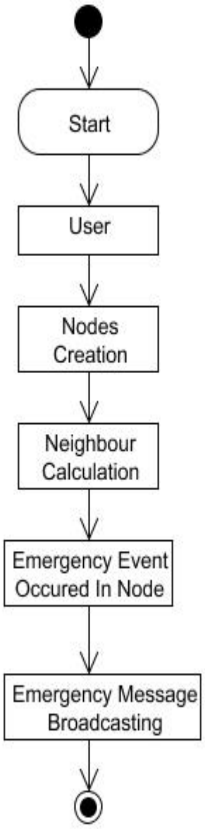

Start
User
Nodes Creation
Neighbour Calculation
Emergency Event Occured In Node
Emergency Message Broadcasting

Figure 4.3: Activity diagram

4.4 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

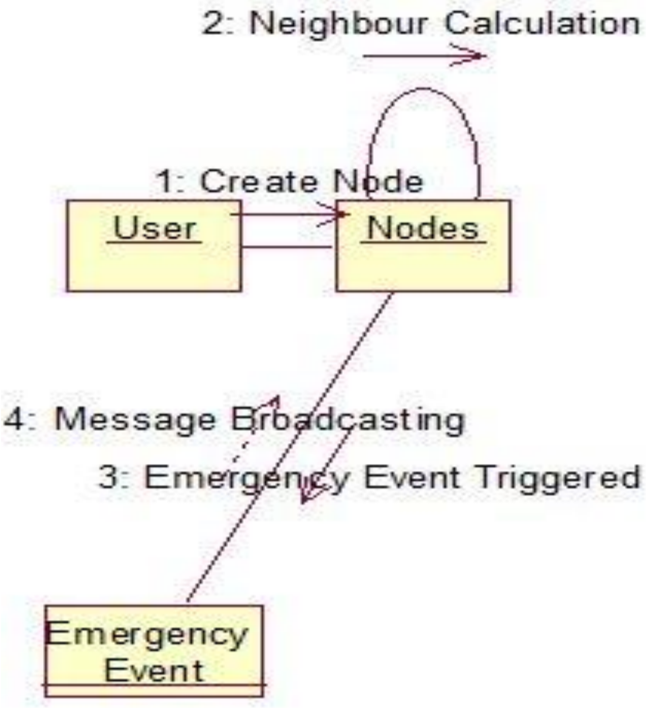


Figure 4.4: Collaboration diagram

4.5 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

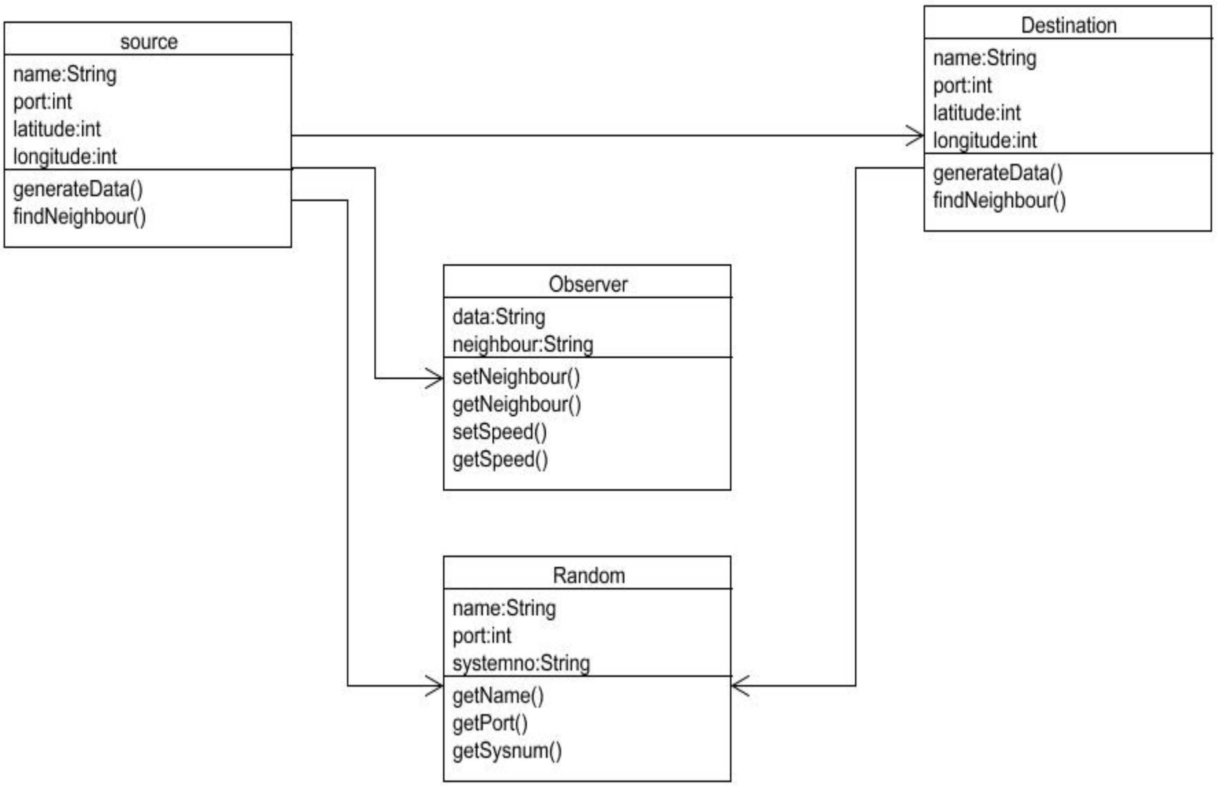


Figure 4.5: Class diagram

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 Architecture overview

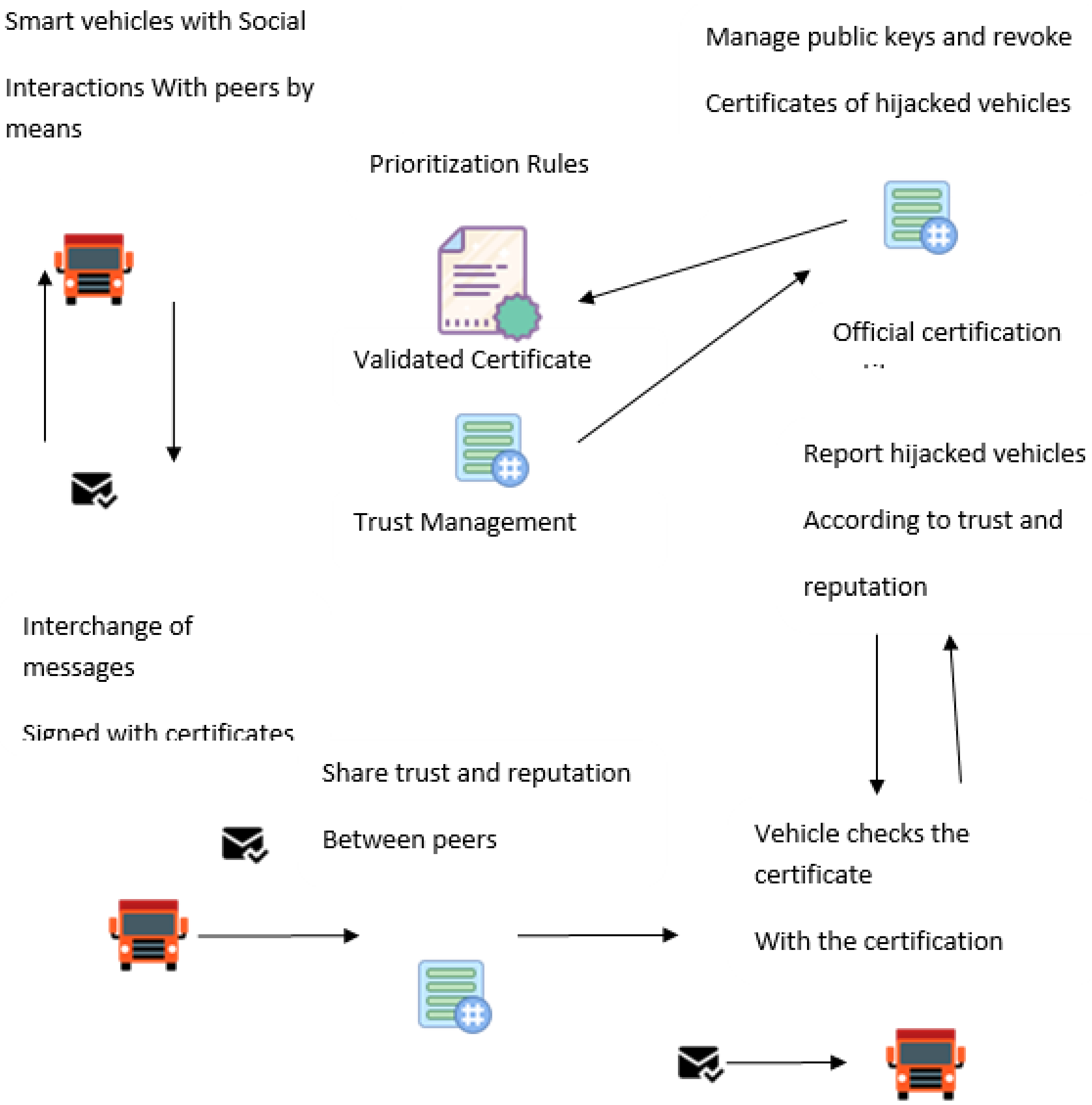


Figure 5.1: System Architecture

5.2 Module design specification

5.2.1 SYSTEM DESIGN

In this module, we create a network consisting of nodes. Each node acts as a vehicle and has its own distance and range. We create nodes by giving latitude and longitude as input which in turn describes the vehicle location. Each node will be dynamic in position that is changing their position dynamically. We create ‘n’ number of nodes based on our requirement to form network environment or network formation. After network formation based on each node latitude and longitude neighbor calculation will be calculated. We calculate neighbor to send messages among nodes and communicate among them. Data will be sent to destination from source via neighbors if both source and destination are not within their communication range. In VANET environment many emergency events can occur like for example Partial Brake, Emergency Brake, and Overtaking etc. When an emergency event occurs it create chaos in the environment. Based on the emergency event occurred an emergency message will be created. After an emergency event occurred an emergency message will be created by the vehicle. Emergency message will be created based on the type of event occurred. Many emergency events may occur in the environment. For example events like sudden break, Partial Brake, Overtaking, ambulance takeover etc are some of the examples of the emergency event. If a vehicle detects a dangerous event, it immediately generates and broadcasts an emergency message to the vehicles in the region of interest (or target region with safety risks), such that the nearby vehicles can take effective actions to avoid traffic accident. In essence, the emergency message, which contains life-critical and time-sensitive information, should be disseminated to all targeted vehicles in a very efficient and effective way.

5.2.2 MODULES

- User registration
- Node creation and Neighbor Calculation
- Emergency Event and traffic calculation
- Blockchain store

MODULE EXPLANATION:

User registration:

Every user has to give their basic details in Registration form, and get signup in our platform, So that all the basic details which the user filled stored in the database, which will be cross checked when we try to sign in.

In the registration form we ask the basic details of their name, password, contact number and email, finally they need to submit and finish the registration form, so that their information will be stored in the database. During sign in ,we ask only their user name and password

Node creation and Neighbor calculation:

In this module, we create a network consisting of nodes. Each node acts as a vehicle and has its own distance and range.

We create nodes by giving latitude and longitude as input which in turn describes the vehicle location. Each node will be dynamic in position that is changing their position dynamically. We create 'n' number of nodes based on our requirement to form network environment or network formation.

Here we create two RSU, which is used to send the user about the traffic status.

For each vehicle, we create a node for each vehicle .There will be port number, which specifies where the vehicle would be , and we created node by calculating latitude and longitude. Latitude ranges from 80 to 200 which is X co-ordinate and longitude ranges from 1000-2000 which is y co-ordinate are used to calculate the distance of neighbor. we created all these nodes in VANET environment, which automatically creates the signal. For the control of vehicle, We have four arrows. If we click left the vehicle moves left and if we click right the vehicle moves right, similarly for top and bottom.

After network formation based on each node latitude and longitude neighbor calculation will be calculated. We calculate neighbor to send messages among nodes and communicate among them. Data will be sent to destination from source via neighbors if both source and destination are not within their communication range.

Emergency Event and Taffic Calculation:

In VANET environment many emergency events can occur like for example Partial Brake, Emergency Brake, and Overtaking etc. When an emergency event occurs it creates chaos in the environment. Based on the emergency event occurred an emergency message will be created.

If we want to know about the status of particular place, select the place using dropdown box appears in the screen. First we need to click request traffic status, the give the place name in traffic update which we want to know. After that request will be sent to RSU, which will reply back by checking their traffic in received text. Here we also set the environment of creating the messages of nearby events in received text to the neighbour.

Broadcasting Emergency Message and block chain store:

After an emergency event occurred an emergency message will be created by the vehicle. Emergency message will be created based on the type of event occurred. Many emergency events may occur in the environment. For example events like sudden break, Partial Brake, Overtaking, ambulance takeover etc are some of the examples of the emergency event. If a vehicle detects a dangerous event, it immediately generates and broadcasts an emergency message to the vehicles in the region of interest (or target region with safety risks), such that the nearby vehicles can take effective actions to avoid traffic accident. In essence, the emergency message, which contains life-critical and time-sensitive information, should be disseminated to all targeted vehicles in a very efficient and effective way.

Emergency event created in VANET environment will be updated in blockchain. The information stored in block chain in encrypted format using MDS algorithm. The information will be decrypted only to the specific user. If we perform attack purposely the message reached neighbour is of corrupted, natural events like sudden break etc…. such events will be sent to neighbour that the sudden break has occurred in the neighbour node in the received text.

5.3 Program design language

5.3.1 JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

5.3.1.1 INTRODUCTION TO JAVA

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

5.3.1.2 WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main,** which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out,** which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things:

- Programming language
- Platform

5.5.1.3 THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented
- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral

- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is unusual in that each Java program is both co implied and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:

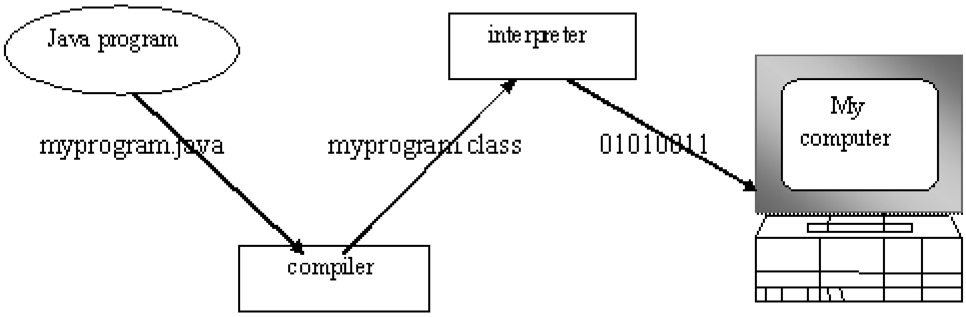


Figure 5.5.1.3: Java compilation

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM).** Every Java interpreter, whether it’s a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can e run on Windows NT, Solaris and Macintos.

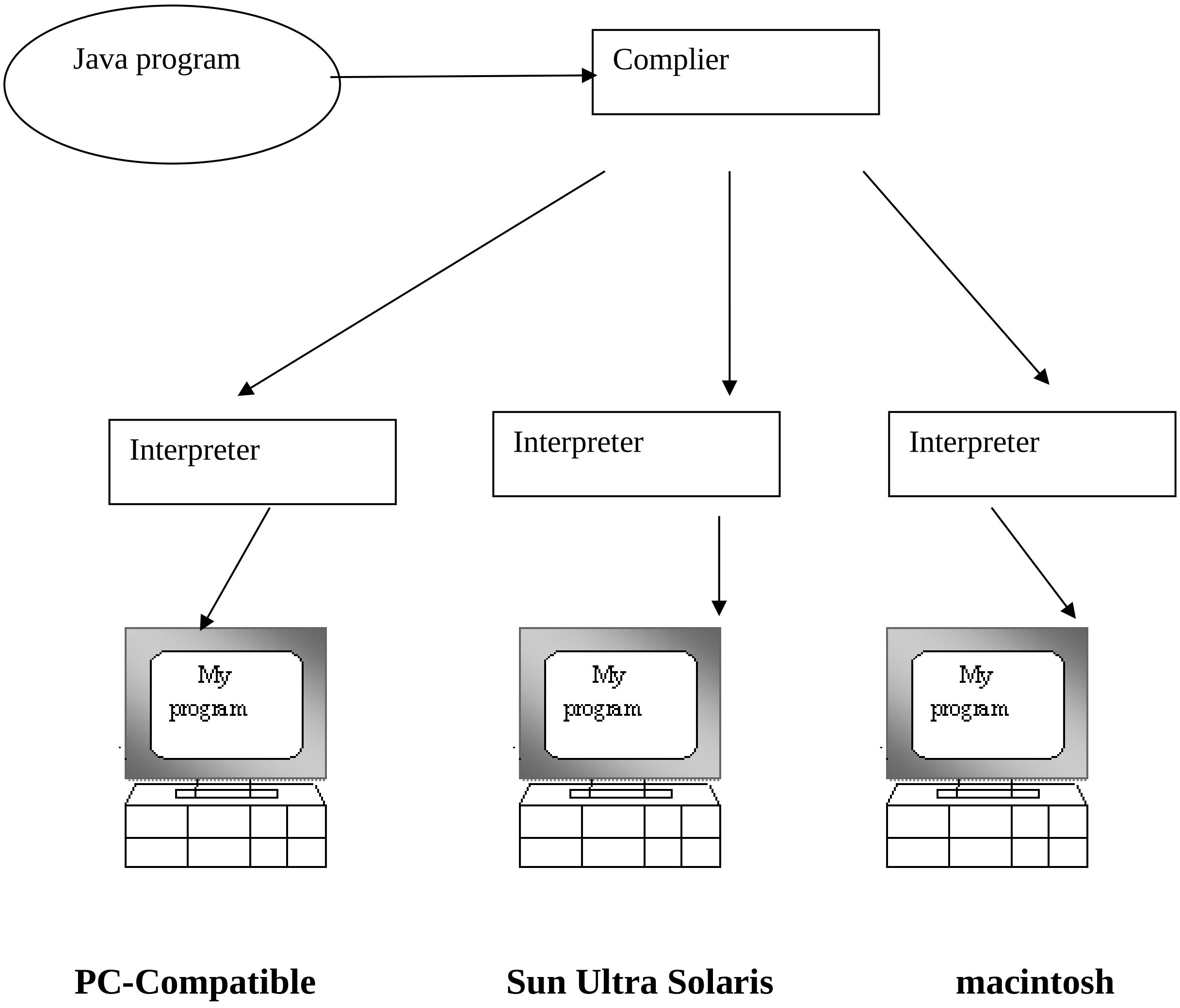


Figure 5.5.1.3: Java virtual machine

5.5.1.4 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components:

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries **(packages)** of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

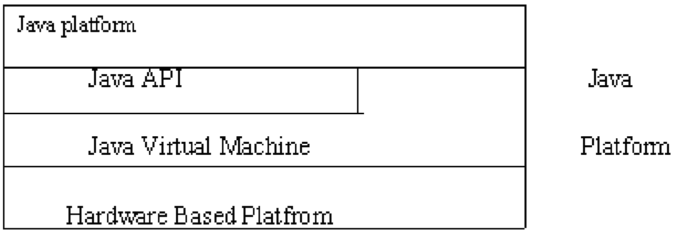


Figure5.5.1.4: Java platform

As a platform-independent environment, Java can be a bit slower than native code. However, smart compliers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

5.5.2 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

Environment

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

Product features

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP paring

History

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

5.3.3 **Introduction to IOT**

IoT refers to an Internet Of Things (IoT). Connecting any device (including everything from cell phones, vehicles, home appliances and other wearable embedded with sensors and actuators) with Internet so that these objects can exchange data with each other on a network. It is interesting to note that there is a difference between IoT and the Internet; it is the absence of Human role. The IoT devices can create information about individual's behaviors, analyze it, and take action (IoT is smarter than Internet: D)

A question would arise in your mind that why we are concerned about IoT? Here is the answer that why you should be concerned about IoT. Say for example you are on your way to a meeting, your car could have access to your calendar and already know the best route to take.If the traffic is heavy your car might send a text to the other party notifying them that you will be late. What if your alarm clock wakes up you at 6 a.m. and then notifies your coffee maker to start making coffee for you? Being able to turn the lights on in your house or heating before coming home using your smartphone? Yes, all these things are possible because of IoT.

Smart System and the Internet of the Things are driven by a combination for:

1. Sensors & Actuators
2. Connectivity
3. People & Process

INTEROPERABILITY IN IOT

- The Internet of Things (IoT) is an incredibly diverse space, encompassing a large variety of hardware form factors and software ecosystems unlike anything we have seen in technology. Smart watches, connected cameras, drones, thermostats, voice-enabled speakers, smart appliances and more—they all live together within the IoT.
- The diversity and innovation that excites many IoT fans is a big challenge not just for manufacturers and developers, but also (and most importantly) consumers. Which technology options should be used when designing or deploying IoT devices? How do they keep up with updated or new operating systems? What about new software and connectivity technologies coming up? Those are just some of today's challenges.

- Having a single, unified communication and software framework for the IoT seems like an ideal solution, but the diverse and fast-paced nature of the IoT makes this utopia a big challenge. Diversity in the IoT is not something to be solved, but an aspect that must be embraced and managed.

Application areas for the Internet of Things

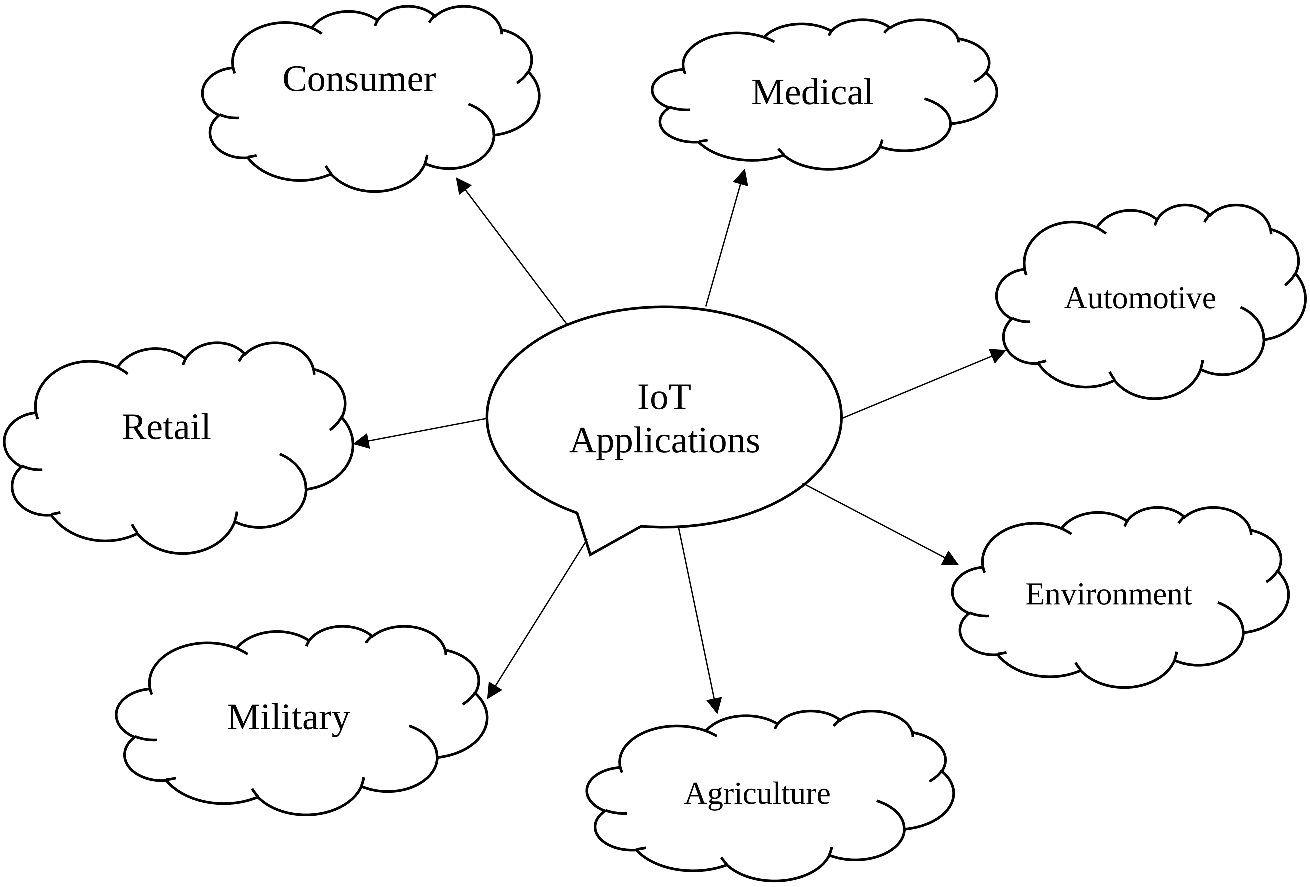


Figure 5.3.3: IoT Applications

Smart Home

The concept of Smart Home is brought up to save time, energy and money. With the introduction of Smart Homes, we would be able to switch on air conditioning before reaching home or switch off lights even after leaving home or unlock the doors to friends for temporary access even when you are not at home.

Smart cities

Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities. IoT will solve major problems faced by the people living in cities like pollution, traffic congestion and shortage of energy supplies etc. By installing sensors and using web applications, citizens can find free available parking slots across the city. Also, the sensors can detect meter tampering issues, general malfunctions and any installation issues in the electricity system.

Wearable's

Wearable devices are installed with sensors and software's which collect data and information about the users. This data is later pre-processed to extract essential insights about user. These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

Healthcare

IoT in healthcare is aimed at empowering people to live healthier life and regular checkup by wearing connected devices. The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.

Challenges to IoT

In the era of IoT, everything is connected, linked up—much more than we see in and around us. IoT is certainly opening door to a lot of opportunities but also to many challenges.

Security Challenges

Security is a big issue with IoT devices. With billions of devices being connected together over Internet, how can people be sure that their information is secure? These security issues can be of the following kinds

Data Encryption

IoT applications collect tons of data. Data retrieval and processing is integral part of the whole IoT environment. Most of this data is personal and needs to be protected through encryption.

Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private. Organizations and individuals use encryption to protect sensitive data stored on computers, servers and mobile devices like phones or tablets.

Data Authentication

After successful encryption of data chances of device itself being hacked still exist. If there is no way to establish the authenticity of the data being communicated to and from an IoT device, security is compromised.

For instance, say you built a temperature sensor for smart homes. Even though you encrypt the data it transfers is there is no way to authenticate the source of data then anyone can make up fake data and send it to your sensor instructing it to cool the room even when its freezing or vice versa.

Side-channel Attacks

Encryption and authentication both in place still leave scope for side channel attacks. Such attacks focus less on the information and more on how that information is being presented. For instance if someone can access data like timing information, power consumption or electromagnetic leak, all of this information can be used for side channel attacks.

Privacy Challenges

Then we have the issue of privacy and data sharing. That is because these devices not only collect personal information like users' names and telephone numbers, but can also monitor user activities (e.g., when users are in their houses and what they had for lunch).

Connectivity Challenges—Billions of devices on a centralized server

One of the biggest challenges for IoT in the future is to connect large number of devices and massive amounts of data that all of these devices are going to produce. There will be need to find out a way to store, track, analyze and make sense of the vast amounts of data that will be generated.

Presently, we rely upon centralized, server/client model to authorize, authenticate, and connect several nodes present on the network. This model is sufficient for the number of IoT devices that are currently a part of the ecosystem.

However, in the future, when hundreds of billions of devices will join the network, it will be difficult to manage all the data. Moreover, the capability of current cloud servers is so less that it can breakdown if it has to handle large amounts of information.

Compatibility and Longevity Challenges-Extra hardware and software

Different technologies like ZigBee, Z-Wave, WI-Fi, Bluetooth and, Bluetooth Low Energy (BTLE) are all battling to become the dominant transport mechanism between devices and hubs. This becomes a major source of problems when a lot of devices have to be connected; such dense connectivity requires the deployment of extra hardware and software.

Conversations about the IoT are taking place all over the world as we are trying to understand how this will impact our lives. We are also trying to understand what the many opportunities and challenges are going to be as more and more devices start to join the IoT. So, all that we can do is educate ourselves about what the IoT is and how it will be after some years.

1.4 Algorithm

- MD5 ALGORITHM is used here.
- Message Digest 5 is widely used in hash function producing 128 bit hash value512 bit input is given as input to the MD5.
- 64 bits get padded to the original 448 bits to get 512 bits, after padding it receives multiple of 512 bits.

- Create MD buffer of 512 bits to store the result of MD5.
- 512 bits of message enter into MD5 and the output will be in 128 bits

1.4.1 Project concern:

- In order to change the normal word to signature , combination of word and number.
- We give the static value of 12345 to MD5 and the final output will be in 128 bits.
- These value get stored in Block chain to reply back to the user.
- Even any of the input the input value changes, whole signature gets changed.
- Hence it is efficient to compare values we use MD5 algorithm.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Coding

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
*** GENERATED FROM TEMPLATE - DO NOT EDIT ***
***       EDIT ../build.xml INSTEAD       ***
-->

<project name="jfx-impl" default="jfx-deployment" basedir=".."
xmlns:j2seproject1="http://www.netbeans.org/ns/j2se-project/1"
         xmlns:j2seproject3="http://www.netbeans.org/ns/j2se-
project/3" xmlns:fx="javafx:com.sun.javafx.tools.ant">
    <description>JavaFX-specific Ant calls</description>


    <!-- Empty placeholders for easier customization in ../build.xml
-->

    <target name="-pre-jfx-jar">
        <!-- Called right before <fx:jar> task. You can override
this target in the ../build.xml file. -->
    </target>

    <target name="-post-jfx-jar">
        <!-- Called right after <fx:jar> task. You can override this
target in the ../build.xml file. -->
    </target>

    <target name="-pre-jfx-deploy">
        <!-- Called right before <fx:deploy> task. You can override
this target in the ../build.xml file. -->
    </target>

    <target name="-post-jfx-deploy">
        <!-- Called right after <fx:deploy> task. You can override
this target in the ../build.xml file. -->
    </target>

    <target name="-pre-jfx-native">
        <!-- Called right before the call to native packager (just
after -pre-jfx-deploy). You can override this target in the
../build.xml file. -->
    </target>

    <target name="-post-jfx-native">
        <!-- Called right after the call to native packager (just
after -post-jfx-deploy). You can override this target in the
../build.xml file. -->
    </target>
    <!-- Check system and JDK version -->

    <target name="-check-operating-system">
        <condition property="running.on.mac">
            <os family="mac"/>
        </condition>
        <condition property="running.on.unix">
            <os family="unix"/>
        </condition>
        <condition property="running.on.windows">
            <os family="windows"/>
        </condition>
        <echo message="running.on.mac = ${running.on.mac}"
level="verbose"/>
        <echo message="running.on.unix = ${running.on.unix}"
level="verbose"/>
        <echo message="running.on.windows = ${running.on.windows}"
level="verbose"/>
    </target>

    <target name="-check-platform-home-fxsdk-java" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.platform.home.fxsdk.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${javafx.sdk}${file.separator}bin${file.separator}java"/>
                    <available
file="${javafx.sdk}${file.separator}bin${file.separator}java.exe"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-fxsdk-java" depends="-check-
platform-home-fxsdk-java" if="do.set.platform.home.fxsdk.java">
        <property name="active.platform.home.java.executable"
value="${javafx.sdk}${file.separator}bin${file.separator}java"/>
    </target>
    <target name="-check-platform-home-java" if="platform.home">
        <condition property="do.set.platform.home.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${platform.home}${file.separator}bin${file.separator}java"/>
                    <available
file="${platform.home}${file.separator}bin${file.separator}java.exe"
/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-java" depends="-set-platform-
home-fxsdk-java,-check-platform-home-java"
if="do.set.platform.home.java">
        <property name="active.platform.home.java.executable"
value="${platform.home}${file.separator}bin${file.separator}java"/>
    </target>
    <target name="-check-platform-home-probjdk-java"
unless="active.platform.home.java.executable">
        <condition property="do.set.platform.home.probjdk.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${java.home}${file.separator}..${file.separator}bin${file.sepa
rator}java"/>
                    <available
file="${java.home}${file.separator}..${file.separator}bin${file.sepa
rator}java.exe"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-probjdk-java" depends="-set-
platform-home-java,-check-platform-home-probjdk-java"
if="do.set.platform.home.probjdk.java">
        <property name="active.platform.home.java.executable"
value="${java.home}${file.separator}..${file.separator}bin${file.sep
arator}java"/>
    </target>
    <target name="-check-platform-home-envjdk-java"
unless="active.platform.home.java.executable">
        <property environment="env"/>
        <condition property="do.set.platform.home.envjdk.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${env.JAVA_HOME}${file.separator}bin${file.separator}java"/>
                    <available
file="${env.JAVA_HOME}${file.separator}bin${file.separator}java.exe"
/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-envjdk-java" depends="-set-
platform-home-probjdk-java,-check-platform-home-envjdk-java"
if="do.set.platform.home.envjdk.java">
        <property environment="env"/>
        <property name="active.platform.home.java.executable"
value="${env.JAVA_HOME}${file.separator}bin${file.separator}java"/>
    </target>
    <target name="-check-platform-home-fxrt-java" depends="-check-
property-javafx.runtime" if="javafx.runtime.defined">
```

```
        <condition property="do.set.platform.home.fxrt.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${javafx.runtime}${file.separator}bin${file.separator}java"/>
                    <available
file="${javafx.runtime}${file.separator}bin${file.separator}java.exe
"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-fxrt-java" depends="-set-
platform-home-envjdk-java,-check-platform-home-fxrt-java"
if="do.set.platform.home.fxrt.java">
        <property name="active.platform.home.java.executable"
value="${javafx.runtime}${file.separator}bin${file.separator}java"/>
        <echo message="Warning: java executable not found in JDK,
evaluating java executable in RT instead." level="info"/>
    </target>
    <target name="-check-platform-home-jre-java"
unless="active.platform.home.java.executable">
        <condition property="do.set.platform.home.jre.java">
            <and>
                <not><isset
property="active.platform.home.java.executable"/></not>
                <or>
                    <available
file="${java.home}${file.separator}bin${file.separator}java"/>
                    <available
file="${java.home}${file.separator}bin${file.separator}java.exe"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-platform-home-jre-java" depends="-set-
platform-home-fxrt-java,-check-platform-home-jre-java"
if="do.set.platform.home.jre.java">
        <property name="active.platform.home.java.executable"
value="${java.home}${file.separator}bin${file.separator}java"/>
        <echo message="Warning: java executable not found in JDK,
evaluating java executable in RT instead." level="info"/>
    </target>
    <target name="-check-platform-home" depends="-set-platform-home-
jre-java">
        <echo message="active.platform.home.java.executable =
${active.platform.home.java.executable}" level="verbose"/>
        <fail message="Error:${line.separator}java executable not
found !" unless="active.platform.home.java.executable"/>
    </target>

    <target name="-check-jdk-version" depends="-do-init,-check-
platform-home" unless="jdk-version-checked-in-jfximpl">
        <local name="version-output"/>
        <exec executable="${active.platform.home.java.executable}"
outputproperty="version-output">
            <arg value="-version"/>
        </exec>
        <echo message="version-output:${line.separator}${version-
output}" level="verbose"/>
        <condition property="have-jdk-older-than-1.6">
            <or>
                <contains string="${version-output}" substring="java
version &quot;1.0"/>
                <contains string="${version-output}" substring="java
version &quot;1.1"/>
                <contains string="${version-output}" substring="java
version &quot;1.2"/>
                <contains string="${version-output}" substring="java
version &quot;1.3"/>
                <contains string="${version-output}" substring="java
version &quot;1.4"/>
                <contains string="${version-output}" substring="java
version &quot;1.5"/>
            </or>
        </condition>
        <fail message="Error:${line.separator}JavaFX 2.0+ projects
require JDK version 1.6+ !" if="have-jdk-older-than-1.6"/>
        <condition property="have-jdk-7u4or5-mac">
            <and>
                <or>
                    <contains string="${version-output}"
substring="java version &quot;1.7.0_04"/>
                    <contains string="${version-output}"
substring="java version &quot;1.7.0_05"/>
                </or>
                <os family="mac"/>
            </and>
        </condition>
        <condition property="have-jdk-pre7u6">
            <or>
                <isset property="have-jdk-older-than-1.6"/>
                <contains string="${version-output}" substring="java
version &quot;1.6"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0&quot;"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_01"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_02"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_03"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_04"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_05"/>
            </or>
        </condition>
        <condition property="have-jdk-pre7u14">
            <or>
```

```
                <isset property="have-jdk-pre7u6"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_06"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_07"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_08"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_09"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_10"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_11"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_12"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_13"/>
            </or>
        </condition>
        <property name="jdk-version-checked-in-jfximpl"
value="true"/>
        <echo message="have-jdk-7u4or5-mac = ${have-jdk-7u4or5-mac}"
level="verbose"/>
        <echo message="have-jdk-pre7u6 = ${have-jdk-pre7u6}"
level="verbose"/>
        <echo message="have-jdk-pre7u14 = ${have-jdk-pre7u14}"
level="verbose"/>
    </target>

    <target name="-check-ant-jre-version" unless="ant-jre-version-
checked-in-jfximpl">
        <local name="version-output"/>
        <exec
executable="${java.home}${file.separator}bin${file.separator}java"
outputproperty="version-output">
            <arg value="-version"/>
        </exec>
        <echo message="version-output:${line.separator}${version-
output}" level="verbose"/>
        <condition property="have-ant-jre-pre7u6">
            <or>
                <contains string="${version-output}" substring="java
version &quot;1.0"/>
                <contains string="${version-output}" substring="java
version &quot;1.1"/>
                <contains string="${version-output}" substring="java
version &quot;1.2"/>
                <contains string="${version-output}" substring="java
version &quot;1.3"/>
                <contains string="${version-output}" substring="java
version &quot;1.4"/>
                <contains string="${version-output}" substring="java
version &quot;1.5"/>
                <contains string="${version-output}" substring="java
version &quot;1.6"/>
```

```
                <contains string="${version-output}" substring="java
version &quot;1.7.0&quot;"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_01"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_02"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_03"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_04"/>
                <contains string="${version-output}" substring="java
version &quot;1.7.0_05"/>
            </or>
        </condition>
        <condition property="have-jdk7-css2bin-bug">
            <!-- as of NB7.4 release date the external css-to-bss
converter is unreliable in all JDK7 versions before 7u40 (with
exception of 7u14)-->
            <and>
                <contains string="${version-output}" substring="java
version &quot;1.7"/>
                <not><matches string="${version-output}"
pattern="\bjava version &quot;1\.7\.0_(14|[4-9].)"/></not>
            </and>
        </condition>
        <property name="ant-jre-version-checked-in-jfximpl"
value="true"/>
        <echo message="have-ant-jre-pre7u6 = ${have-ant-jre-pre7u6}"
level="verbose"/>
        <echo message="have-jdk7-css2bin-bug = ${have-jdk7-css2bin-
bug}" level="verbose"/>
    </target>

    <target name="-check-jdk-7u4or5-mac" depends="-check-jdk-
version" if="have-jdk-7u4or5-mac">
        <fail message="Error:${line.separator}JDK 7u4 Mac and 7u5
Mac do not support WebStart and JavaFX 2.0+ browser plugin
technologies.${line.separator}Please upgrade to JDK 7u6 or later."/>
    </target>


    <!-- Check availability of JavaFX SDK deployment support (ant-
javafx.jar) -->

    <target name="-check-endorsed-javafx-ant-classpath">
        <condition property="endorsed-javafx-ant-classpath-
available">
            <and>
                <isset property="endorsed.javafx.ant.classpath"/>
                <not>
                    <equals arg1="${endorsed.javafx.ant.classpath}"
arg2=""/>
                </not>
            </and>
        </condition>
        <echo message="endorsed-javafx-ant-classpath-available =
${endorsed-javafx-ant-classpath-available}" level="verbose"/>
    </target>

    <target name="-check-property-javafx.sdk">
        <echo message="javafx.sdk = ${javafx.sdk}" level="verbose"/>
        <condition property="javafx.sdk.defined">
            <and>
                <isset property="javafx.sdk"/>
                <not><contains string="${javafx.sdk}"
substring="$${platform" casesensitive="false"/></not>
            </and>
        </condition>
        <condition property="javafx.sdk.missing+default">
            <and>
                <equals arg1="${platform.active}"
arg2="Default_JavaFX_Platform" trim="true"/>
                <not><isset property="javafx.sdk.defined"/></not>
            </and>
        </condition>
        <condition property="javafx.sdk.missing-default">
            <and>
                <not><equals arg1="${platform.active}"
arg2="Default_JavaFX_Platform" trim="true"/></not>
                <not><isset property="javafx.sdk.defined"/></not>
            </and>
        </condition>
        <echo message="javafx.sdk.defined = ${javafx.sdk.defined}"
level="verbose"/>
        <echo message="javafx.sdk.missing+default =
${javafx.sdk.missing+default}" level="verbose"/>
        <echo message="javafx.sdk.missing-default =
${javafx.sdk.missing-default}" level="verbose"/>
    </target>

    <target name="-check-ant-javafx-in-fxsdk-lib" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.ant-javafx.in.fxsdk.lib">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}lib${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-fxsdk-lib" depends="-check-ant-
javafx-in-fxsdk-lib" if="do.set.ant-javafx.in.fxsdk.lib">
        <property name="ant-javafx.jar.location"
value="${javafx.sdk}${file.separator}lib${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-fxsdk-tools" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.ant-javafx.in.fxsdk.tools">
```

```
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}tools${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-fxsdk-tools" depends="-set-ant-
javafx-in-fxsdk-lib,-check-ant-javafx-in-fxsdk-tools"
if="do.set.ant-javafx.in.fxsdk.tools">
        <property name="ant-javafx.jar.location"
value="${javafx.sdk}${file.separator}tools${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-platform-home-lib"
if="platform.home">
        <condition property="do.set.ant-
javafx.in.platform.home.lib">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${platform.home}${file.separator}lib${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-platform-home-lib" depends="-
set-ant-javafx-in-fxsdk-tools,-check-ant-javafx-in-platform-home-
lib" if="do.set.ant-javafx.in.platform.home.lib">
        <property name="ant-javafx.jar.location"
value="${platform.home}${file.separator}lib${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-platform-home-tools"
if="platform.home">
        <condition property="do.set.ant-
javafx.in.platform.home.tools">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${platform.home}${file.separator}tools${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-platform-home-tools" depends="-
set-ant-javafx-in-platform-home-lib,-check-ant-javafx-in-platform-
home-tools" if="do.set.ant-javafx.in.platform.home.tools">
        <property name="ant-javafx.jar.location"
value="${platform.home}${file.separator}tools${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-probjdk-lib" unless="ant-
javafx.jar.location">
        <condition property="do.set.ant-javafx.in.probjdk.lib">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${java.home}${file.separator}..${file.separator}lib${file.sepa
rator}ant-javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-probjdk-lib" depends="-set-ant-
javafx-in-platform-home-tools,-check-ant-javafx-in-probjdk-lib"
if="do.set.ant-javafx.in.probjdk.lib">
        <property name="ant-javafx.jar.location"
value="${java.home}${file.separator}..${file.separator}lib${file.sep
arator}ant-javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-probjdk-tools" unless="ant-
javafx.jar.location">
        <condition property="do.set.ant-javafx.in.probjdk.tools">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${java.home}${file.separator}..${file.separator}tools${file.se
parator}ant-javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-probjdk-tools" depends="-set-
ant-javafx-in-probjdk-lib,-check-ant-javafx-in-probjdk-tools"
if="do.set.ant-javafx.in.probjdk.tools">
        <property name="ant-javafx.jar.location"
value="${java.home}${file.separator}..${file.separator}tools${file.s
eparator}ant-javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-macjdk-lib" unless="ant-
javafx.jar.location">
        <condition property="do.set.ant-javafx.in.macjdk.lib">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${java.home}${file.separator}lib${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-macjdk-lib" depends="-set-ant-
javafx-in-probjdk-tools,-check-ant-javafx-in-macjdk-lib"
if="do.set.ant-javafx.in.macjdk.lib">
        <property name="ant-javafx.jar.location"
value="${java.home}${file.separator}lib${file.separator}ant-
javafx.jar"/>
```

```
    </target>
    <target name="-check-ant-javafx-in-envjdk-lib" unless="ant-
javafx.jar.location">
        <property environment="env"/>
        <condition property="do.set.ant-javafx.in.envjdk.lib">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${env.JAVA_HOME}${file.separator}lib${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-envjdk-lib" depends="-set-ant-
javafx-in-macjdk-lib,-check-ant-javafx-in-envjdk-lib"
if="do.set.ant-javafx.in.envjdk.lib">
        <property name="ant-javafx.jar.location"
value="${env.JAVA_HOME}${file.separator}lib${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-check-ant-javafx-in-envjdk-tools" unless="ant-
javafx.jar.location">
        <property environment="env"/>
        <condition property="do.set.ant-javafx.in.envjdk.tools">
            <and>
                <not><isset property="ant-
javafx.jar.location"/></not>
                <available
file="${env.JAVA_HOME}${file.separator}tools${file.separator}ant-
javafx.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-ant-javafx-in-envjdk-tools" depends="-set-
ant-javafx-in-envjdk-lib,-check-ant-javafx-in-envjdk-tools"
if="do.set.ant-javafx.in.envjdk.tools">
        <property name="ant-javafx.jar.location"
value="${env.JAVA_HOME}${file.separator}tools${file.separator}ant-
javafx.jar"/>
    </target>
    <target name="-pre-check-ant-javafx-version" depends="-set-ant-
javafx-in-envjdk-tools" unless="ant-javafx-version-already-checked-
in-jfximpl">
        <condition property="do.check.ant-javafx.version">
            <and>
                <isset property="ant-javafx.jar.location"/>
                <not><isset property="ant-javafx-version-already-
checked-in-jfximpl"/></not>
            </and>
        </condition>
    </target>
    <target name="-set-endorsed-javafx-ant-classpath" depends="-
check-endorsed-javafx-ant-classpath,-pre-check-ant-javafx-version"
if="endorsed-javafx-ant-classpath-available">
```

```
        <property name="javafx.ant.classpath"
value="${endorsed.javafx.ant.classpath}:${ant-
javafx.jar.location}"/>
    </target>
    <target name="-set-javafx-ant-classpath" depends="-check-
endorsed-javafx-ant-classpath,-pre-check-ant-javafx-version"
unless="endorsed-javafx-ant-classpath-available">
        <property name="javafx.ant.classpath" value="${ant-
javafx.jar.location}"/>
    </target>
    <target name="-check-ant-javafx-version" depends="-pre-check-
ant-javafx-version,
            -set-endorsed-javafx-ant-classpath,-set-javafx-ant-
classpath" if="do.check.ant-javafx.version">
        <echo message="ant-javafx.jar.location = ${ant-
javafx.jar.location}" level="verbose"/>
        <echo message="javafx.ant.classpath =
${javafx.ant.classpath}" level="verbose"/>
        <taskdef resource="com/sun/javafx/tools/ant/antlib.xml"
            uri="javafx:com.sun.javafx.tools.ant"
            classpath="${javafx.ant.classpath}"/>
        <condition property="have-fx-ant-init">
            <typefound name="javafx:com.sun.javafx.tools.ant:init-
ant"/>
        </condition>
        <property name="ant-javafx-version-already-checked-in-
jfximpl" value="true"/>
        <echo message="have-fx-ant-init = ${have-fx-ant-init}"
level="verbose"/>
    </target>
    <target name="-check-jfx-sdk-version-old" depends="-check-ant-
javafx-version" unless="have-fx-ant-init">
        <property name="javafx.ant.version" value="1.0"/>
    </target>
    <target name="-check-jfx-sdk-version-new" depends="-check-ant-
javafx-version" if="have-fx-ant-init">
        <fx:init-ant/>
        <condition property="have-fx-ant-api-1.1">
            <!-- new features from JavaFX 2.0.2 are available in API
version 1.1 or later -->
            <matches pattern="1.[1-9]"
string="${javafx.ant.version}"/>
        </condition>
        <condition property="have-fx-ant-api-1.2">
            <!-- new features from JavaFX 2.2 are available in API
version 1.2 or later -->
            <matches pattern="1.[2-9]"
string="${javafx.ant.version}"/>
        </condition>
    </target>
    <target name="-check-jfx-sdk-version" depends="-check-jfx-sdk-
version-old, -check-jfx-sdk-version-new"
unless="jfx.sdk.version.checked">
        <echo message="Detected JavaFX Ant API version
${javafx.ant.version}" level="info"/>
```

```
        <echo message="have-fx-ant-api-1.1 = ${have-fx-ant-api-1.1}" 
level="verbose"/>
        <echo message="have-fx-ant-api-1.2 = ${have-fx-ant-api-1.2}" 
level="verbose"/>
        <echo message="javafx.ant.classpath = 
${javafx.ant.classpath}" level="verbose"/>
        <property name="jfx.sdk.version.checked" value="true"/>
    </target>

    <target name="-check-jfx-deployment" depends="-check-jdk-
version,-check-jfx-sdk-version">
        <condition property="jfx-deployment-available">
            <and>
                <or>
                    <isset property="do.set.ant-
javafx.in.fxsdk.lib"/>
                    <isset property="do.set.ant-
javafx.in.fxsdk.tools"/>
                    <isset property="do.set.ant-
javafx.in.platform.home.lib"/>
                    <isset property="do.set.ant-
javafx.in.platform.home.tools"/>
                    <isset property="do.set.ant-
javafx.in.probjdk.lib"/>
                    <isset property="do.set.ant-
javafx.in.probjdk.tools"/>
                    <isset property="do.set.ant-
javafx.in.envjdk.lib"/>
                    <isset property="do.set.ant-
javafx.in.envjdk.tools"/>
                </or>
                <isset property="ant-javafx.jar.location"/>
            </and>
        </condition>
        <condition property="jfx-deployment-missing+jdk7u6">
            <and>
                <not><isset property="jfx-deployment-
available"/></not>
                <not><isset property="have-jdk-pre7u6"/></not>
            </and>
        </condition>
        <condition property="jfx-deployment-
missing+javafx.sdk.missing+default">
            <and>
                <not><isset property="jfx-deployment-
available"/></not>
                <isset property="have-jdk-pre7u6"/>
                <isset property="javafx.sdk.missing+default"/>
            </and>
        </condition>
        <condition property="jfx-deployment-
missing+javafx.sdk.missing-default">
            <and>
                <not><isset property="jfx-deployment-
available"/></not>
                <isset property="have-jdk-pre7u6"/>
                <isset property="javafx.sdk.missing-default"/>
            </and>
        </condition>
        <fail message="Error:${line.separator}JavaFX deployment
library not found in active JDK.${line.separator}Please check that
the JDK is correctly installed and its version is at least 7u4 on
Mac or 7u6 on other systems." if="jfx-deployment-missing+jdk7u6"/>
        <fail message="Error:${line.separator}JavaFX deployment
library not found.${line.separator}JavaFX SDK path undefined. Check
the definition of ${platform.active} in Java Platform
Manager${line.separator}(or directly the properties platform.active
and javafx.sdk in project.properties file).${line.separator}Note: If
missing, the default JavaFX-enabled platform gets created
automatically when creating a new JavaFX Project." if="jfx-
deployment-missing+javafx.sdk.missing+default"/>
        <fail message="Error:${line.separator}JavaFX deployment
library not found.${line.separator}JavaFX SDK path undefined. Check
the definition of ${platform.active} in Java Platform
Manager${line.separator}(or directly the properties platform.active
and javafx.sdk in project.properties file)." if="jfx-deployment-
missing+javafx.sdk.missing-default"/>
        <fail message="Error:${line.separator}JavaFX deployment
library not found." unless="jfx-deployment-available"/>
        <echo message="jfx-deployment-available = ${jfx-deployment-
available}" level="verbose"/>
    </target>


    <!-- Check availability of main JavaFX runtime jar (jfxrt.jar) -
->

    <target name="-check-property-javafx.runtime">
        <echo message="javafx.runtime = ${javafx.runtime}"
level="verbose"/>
        <condition property="javafx.runtime.defined">
            <and>
                <isset property="javafx.runtime"/>
                <not><contains string="${javafx.runtime}"
substring="$${platform" casesensitive="false"/></not>
            </and>
        </condition>
        <condition property="javafx.runtime.missing+default">
            <and>
                <equals arg1="${platform.active}"
arg2="Default_JavaFX_Platform" trim="true"/>
                <not><isset
property="javafx.runtime.defined"/></not>
            </and>
        </condition>
        <condition property="javafx.runtime.missing-default">
            <and>
                <not><equals arg1="${platform.active}"
arg2="Default_JavaFX_Platform" trim="true"/></not>
                <not><isset
property="javafx.runtime.defined"/></not>
            </and>
        </condition>
        <echo message="javafx.runtime.defined =
${javafx.runtime.defined}" level="verbose"/>
        <echo message="javafx.runtime.missing+default =
${javafx.runtime.missing+default}" level="verbose"/>
        <echo message="javafx.runtime.missing-default =
${javafx.runtime.missing-default}" level="verbose"/>
    </target>

    <target name="-check-jfxrt-in-fxrt" depends="-check-property-
javafx.runtime" if="javafx.runtime.defined">
        <condition property="do.set.jfxrt.in.fxrt.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.runtime}${file.separator}lib${file.separator}jfxrt.ja
r"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.fxrt.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.fxrt.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.runtime}${file.separator}lib${file.separator}ext${fil
e.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-fxrt-old" depends="-check-jfxrt-in-
fxrt" if="do.set.jfxrt.in.fxrt.old">
        <property name="jfxrt.jar.location"
value="${javafx.runtime}${file.separator}lib${file.separator}jfxrt.j
ar"/>
    </target>
    <target name="-set-jfxrt-in-fxrt-new" depends="-set-jfxrt-in-
fxrt-old,-check-jfxrt-in-fxrt" if="do.set.jfxrt.in.fxrt.new">
        <property name="jfxrt.jar.location"
value="${javafx.runtime}${file.separator}lib${file.separator}ext${fi
le.separator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-fxsdk-jre" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.jfxrt.in.fxsdk.jre.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}jre${file.separator}lib${file.se
parator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.fxsdk.jre.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.fxsdk.jre.old"/></not>
```

```
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}jre${file.separator}lib${file.se
parator}ext${file.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-fxsdk-jre-old" depends="-set-jfxrt-
in-fxrt-new,-check-jfxrt-in-fxsdk-jre"
if="do.set.jfxrt.in.fxsdk.jre.old">
        <property name="jfxrt.jar.location"
value="${javafx.sdk}${file.separator}jre${file.separator}lib${file.s
eparator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-fxsdk-jre-new" depends="-set-jfxrt-
in-fxsdk-jre-old,-check-jfxrt-in-fxsdk-jre"
if="do.set.jfxrt.in.fxsdk.jre.new">
        <property name="jfxrt.jar.location"
value="${javafx.sdk}${file.separator}jre${file.separator}lib${file.s
eparator}ext${file.separator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-fxsdk-rt" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.jfxrt.in.fxsdk.rt.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}rt${file.separator}lib${file.sep
arator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.fxsdk.rt.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.fxsdk.rt.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${javafx.sdk}${file.separator}rt${file.separator}lib${file.sep
arator}ext${file.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-fxsdk-rt-old" depends="-set-jfxrt-
in-fxsdk-jre-new,-check-jfxrt-in-fxsdk-rt"
if="do.set.jfxrt.in.fxsdk.rt.old">
        <property name="jfxrt.jar.location"
value="${javafx.sdk}${file.separator}rt${file.separator}lib${file.se
parator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-fxsdk-rt-new" depends="-set-jfxrt-
in-fxsdk-rt-old,-check-jfxrt-in-fxsdk-rt"
if="do.set.jfxrt.in.fxsdk.rt.new">
        <property name="jfxrt.jar.location"
value="${javafx.sdk}${file.separator}rt${file.separator}lib${file.se
parator}ext${file.separator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-platform-home-jre"
if="platform.home">
        <condition property="do.set.jfxrt.in.platform.home.jre.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${platform.home}${file.separator}jre${file.separator}lib${file
.separator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.platform.home.jre.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.platform.home.jre.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${platform.home}${file.separator}jre${file.separator}lib${file
.separator}ext${file.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-platform-home-jre-old" depends="-
set-jfxrt-in-fxsdk-rt-new,-check-jfxrt-in-platform-home-jre"
if="do.set.jfxrt.in.platform.home.jre.old">
        <property name="jfxrt.jar.location"
value="${platform.home}${file.separator}jre${file.separator}lib${fil
e.separator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-platform-home-jre-new" depends="-
set-jfxrt-in-platform-home-jre-old,-check-jfxrt-in-platform-home-
jre" if="do.set.jfxrt.in.platform.home.jre.new">
        <property name="jfxrt.jar.location"
value="${platform.home}${file.separator}jre${file.separator}lib${fil
e.separator}ext${file.separator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-platform-home-rt"
if="platform.home">
        <condition property="do.set.jfxrt.in.platform.home.rt.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${platform.home}${file.separator}rt${file.separator}lib${file.
separator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.platform.home.rt.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.platform.home.rt.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${platform.home}${file.separator}rt${file.separator}lib${file.
separator}ext${file.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-platform-home-rt-old" depends="-set-
jfxrt-in-platform-home-jre-new,-check-jfxrt-in-platform-home-rt"
if="do.set.jfxrt.in.platform.home.rt.old">
        <property name="jfxrt.jar.location"
value="${platform.home}${file.separator}rt${file.separator}lib${file
.separator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-platform-home-rt-new" depends="-set-
jfxrt-in-platform-home-rt-old,-check-jfxrt-in-platform-home-rt"
if="do.set.jfxrt.in.platform.home.rt.new">
        <property name="jfxrt.jar.location"
value="${platform.home}${file.separator}rt${file.separator}lib${file
.separator}ext${file.separator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-jre" unless="jfxrt.jar.location">
        <condition property="do.set.jfxrt.in.jre.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${java.home}${file.separator}lib${file.separator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.jre.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.jre.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${java.home}${file.separator}lib${file.separator}ext${file.sep
arator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-jre-old" depends="-set-jfxrt-in-
platform-home-rt-new,-check-jfxrt-in-jre"
if="do.set.jfxrt.in.jre.old">
        <property name="jfxrt.jar.location"
value="${java.home}${file.separator}lib${file.separator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-jre-new" depends="-set-jfxrt-in-jre-
old,-check-jfxrt-in-jre" if="do.set.jfxrt.in.jre.new">
        <property name="jfxrt.jar.location"
value="${java.home}${file.separator}lib${file.separator}ext${file.se
parator}jfxrt.jar"/>
    </target>
    <target name="-check-jfxrt-in-envjdk-jre"
unless="jfxrt.jar.location">
        <property environment="env"/>
        <condition property="do.set.jfxrt.in.envjdk.jre.old">
            <and>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${env.JAVA_HOME}${file.separator}jre${file.separator}lib${file
.separator}jfxrt.jar"/>
            </and>
        </condition>
        <condition property="do.set.jfxrt.in.envjdk.jre.new">
            <and>
                <not><isset
property="do.set.jfxrt.in.envjdk.jre.old"/></not>
                <not><isset property="jfxrt.jar.location"/></not>
                <available
file="${env.JAVA_HOME}${file.separator}jre${file.separator}lib${file
.separator}ext${file.separator}jfxrt.jar"/>
            </and>
        </condition>
    </target>
    <target name="-set-jfxrt-in-envjdk-jre-old" depends="-set-jfxrt-
in-jre-new,-check-jfxrt-in-envjdk-jre"
if="do.set.jfxrt.in.envjdk.jre.old">
        <property name="jfxrt.jar.location"
value="${env.JAVA_HOME}${file.separator}jre${file.separator}lib${fil
e.separator}jfxrt.jar"/>
    </target>
    <target name="-set-jfxrt-in-envjdk-jre-new" depends="-set-jfxrt-
in-envjdk-jre-old,-check-jfxrt-in-envjdk-jre"
if="do.set.jfxrt.in.envjdk.jre.new">
        <property name="jfxrt.jar.location"
value="${env.JAVA_HOME}${file.separator}jre${file.separator}lib${fil
e.separator}ext${file.separator}jfxrt.jar"/>
    </target>
    <target name="-pre-check-jfx-runtime" depends="-set-jfxrt-in-
envjdk-jre-new">
        <echo message="jfxrt.jar.location = ${jfxrt.jar.location}"
level="verbose"/>
    </target>

    <target name="-check-jfx-runtime" depends="-check-jdk-version, -
pre-check-jfx-runtime">
        <condition property="jfx-runtime-available">
            <and>
                <or>
                    <isset property="do.set.jfxrt.in.fxrt.old"/>
                    <isset property="do.set.jfxrt.in.fxrt.new"/>
                    <isset
property="do.set.jfxrt.in.fxsdk.jre.old"/>
                    <isset
property="do.set.jfxrt.in.fxsdk.jre.new"/>
                    <isset property="do.set.jfxrt.in.fxsdk.rt.old"/>
                    <isset property="do.set.jfxrt.in.fxsdk.rt.new"/>
                    <isset
property="do.set.jfxrt.in.platform.home.jre.old"/>
                    <isset
property="do.set.jfxrt.in.platform.home.jre.new"/>
                    <isset
property="do.set.jfxrt.in.platform.home.rt.old"/>
                    <isset
property="do.set.jfxrt.in.platform.home.rt.new"/>
                    <isset property="do.set.jfxrt.in.jre.old"/>
                    <isset property="do.set.jfxrt.in.jre.new"/>
                    <isset
property="do.set.jfxrt.in.envjdk.jre.old"/>
```

```
                    <isset
property="do.set.jfxrt.in.envjdk.jre.new"/>
                </or>
                <isset property="jfxrt.jar.location"/>
            </and>
        </condition>
        <fail message="Error:${line.separator}JavaFX runtime JAR not
found." unless="jfx-runtime-available"/>
        <echo message="jfx-runtime-available = ${jfx-runtime-
available}" level="verbose"/>
    </target>


    <!-- Check availability of WebStart executable -->

    <target name="-check-webstart-in-fxrt" depends="-check-property-
javafx.runtime" if="javafx.runtime.defined">
        <condition property="do.set.webstart.in.fxrt">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <isset property="javafx.runtime.defined"/>
                <or>
                    <available
file="${javafx.runtime}${file.separator}bin${file.separator}javaws.e
xe"/>
                    <available
file="${javafx.runtime}${file.separator}bin${file.separator}javaws"/
>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-fxrt" depends="-check-webstart-
in-fxrt" if="do.set.webstart.in.fxrt">
        <property name="active.webstart.executable"
value="${javafx.runtime}${file.separator}bin${file.separator}javaws"
/>
    </target>
    <target name="-check-webstart-in-fxsdk-jre" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.webstart.in.fxsdk.jre">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <isset property="javafx.sdk.defined"/>
                <or>
                    <available
file="${javafx.sdk}${file.separator}jre${file.separator}bin${file.se
parator}javaws.exe"/>
                    <available
file="${javafx.sdk}${file.separator}jre${file.separator}bin${file.se
parator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-fxsdk-jre" depends="-set-
webstart-in-fxrt,-check-webstart-in-fxsdk-jre"
if="do.set.webstart.in.fxsdk.jre">
        <property name="active.webstart.executable"
value="${javafx.sdk}${file.separator}jre${file.separator}bin${file.s
eparator}javaws"/>
    </target>
    <target name="-check-webstart-in-fxsdk" depends="-check-
property-javafx.sdk" if="javafx.sdk.defined">
        <condition property="do.set.webstart.in.fxsdk">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <isset property="javafx.sdk.defined"/>
                <or>
                    <available
file="${javafx.sdk}${file.separator}bin${file.separator}javaws.exe"/
>
                    <available
file="${javafx.sdk}${file.separator}bin${file.separator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-fxsdk" depends="-set-webstart-in-
fxsdk-jre,-check-webstart-in-fxsdk" if="do.set.webstart.in.fxsdk">
        <property name="active.webstart.executable"
value="${javafx.sdk}${file.separator}bin${file.separator}javaws"/>
    </target>
    <target name="-check-webstart-in-platform-home-jre"
if="platform.home">
        <condition property="do.set.webstart.in.platform.home.jre">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <or>
                    <available
file="${platform.home}${file.separator}jre${file.separator}bin${file
.separator}javaws.exe"/>
                    <available
file="${platform.home}${file.separator}jre${file.separator}bin${file
.separator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-platform-home-jre" depends="-set-
webstart-in-fxsdk,-check-webstart-in-platform-home-jre"
if="do.set.webstart.in.platform.home.jre">
        <property name="active.webstart.executable"
value="${platform.home}${file.separator}jre${file.separator}bin${fil
e.separator}javaws"/>
    </target>
    <target name="-check-webstart-in-platform-home"
if="platform.home">
```

```
        <condition property="do.set.webstart.in.platform.home">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <or>
                    <available
file="${platform.home}${file.separator}bin${file.separator}javaws.ex
e"/>
                    <available
file="${platform.home}${file.separator}bin${file.separator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-platform-home" depends="-set-
webstart-in-platform-home-jre,-check-webstart-in-platform-home"
if="do.set.webstart.in.platform.home">
        <property name="active.webstart.executable"
value="${platform.home}${file.separator}bin${file.separator}javaws"/
>
    </target>
    <target name="-check-webstart-in-jre"
unless="active.webstart.executable">
        <condition property="do.set.webstart.in.jre">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <or>
                    <available
file="${java.home}${file.separator}bin${file.separator}javaws.exe"/>
                    <available
file="${java.home}${file.separator}bin${file.separator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-jre" depends="-set-webstart-in-
platform-home,-check-webstart-in-jre" if="do.set.webstart.in.jre">
        <property name="active.webstart.executable"
value="${java.home}${file.separator}bin${file.separator}javaws"/>
    </target>
    <target name="-check-webstart-in-probjdk"
unless="active.webstart.executable">
        <condition property="do.set.webstart.in.probjdk">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <or>
                    <available
file="${java.home}${file.separator}..${file.separator}bin${file.sepa
rator}javaws.exe"/>
                    <available
file="${java.home}${file.separator}..${file.separator}bin${file.sepa
rator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-probjdk" depends="-set-webstart-
in-jre,-check-webstart-in-probjdk" if="do.set.webstart.in.probjdk">
        <property name="active.webstart.executable"
value="${java.home}${file.separator}..${file.separator}bin${file.sep
arator}javaws"/>
    </target>
    <target name="-check-webstart-in-envjdk"
unless="active.webstart.executable">
        <property environment="env"/>
        <condition property="do.set.webstart.in.envjdk">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <or>
                    <available
file="${env.JAVA_HOME}${file.separator}bin${file.separator}javaws.ex
e"/>
                    <available
file="${env.JAVA_HOME}${file.separator}bin${file.separator}javaws"/>
                </or>
            </and>
        </condition>
    </target>
    <target name="-set-webstart-in-envjdk" depends="-set-webstart-
in-probjdk,-check-webstart-in-envjdk"
if="do.set.webstart.in.envjdk">
        <property name="active.webstart.executable"
value="${env.JAVA_HOME}${file.separator}bin${file.separator}javaws"/
>
    </target>
    <target name="-pre-check-webstart-in-unix" depends="-check-
operating-system,-set-webstart-in-envjdk"
unless="active.webstart.executable">
        <condition property="running.on.unix-
active.webstart.executable">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
                <isset property="running.on.unix"/>
            </and>
        </condition>
    </target>
    <target name="-check-webstart-in-unix" depends="-pre-check-
webstart-in-unix" if="running.on.unix-active.webstart.executable">
        <local name="exec.which.javaws.result"/>
        <exec executable="command" failifexecutionfails="false"
failonerror="false" resultproperty="exec.which.javaws.result"
outputproperty="exec.which.javaws.output">
            <arg line="-v javaws"/>
        </exec>
        <condition property="do.set.webstart.in.unix">
            <and>
                <not><isset
property="active.webstart.executable"/></not>
```

```
                <isset property="exec.which.javaws.result"/>
                <equals arg1="${exec.which.javaws.result}"
arg2="0"/>
                <isset property="exec.which.javaws.output"/>
                <not><equals arg1="${exec.which.javaws.output}"
arg2=""/></not>
            </and>
        </condition>
        <echo message="do.set.webstart.in.unix =
${do.set.webstart.in.unix}" level="verbose"/>
    </target>
    <target name="-set-webstart-in-unix" depends="-set-webstart-in-
envjdk,-check-webstart-in-unix" if="do.set.webstart.in.unix">
        <property name="active.webstart.executable"
value="${exec.which.javaws.output}"/>
    </target>
    <target name="-pre-check-jfx-webstart" depends="-set-webstart-
in-unix">
        <echo message="active.webstart.executable =
${active.webstart.executable}" level="verbose"/>
    </target>

    <target name="-check-jfx-webstart" depends="-pre-check-jfx-
webstart">
        <condition property="jfx-webstart-available">
            <and>
                <or>
                    <isset property="do.set.webstart.in.fxrt"/>
                    <isset property="do.set.webstart.in.fxsdk.jre"/>
                    <isset property="do.set.webstart.in.fxsdk"/>
                    <isset
property="do.set.webstart.in.platform.home.jre"/>
                    <isset
property="do.set.webstart.in.platform.home"/>
                    <isset property="do.set.webstart.in.jre"/>
                    <isset property="do.set.webstart.in.probjdk"/>
                    <isset property="do.set.webstart.in.envjdk"/>
                    <isset property="do.set.webstart.in.unix"/>
                </or>
        </exec>
    </target>

    <target name="jfxbe-profile" if="profiler.configured"
        depends="-profile-check-html"
        description="Profile JavaFX project in browser">
        <startprofiler/>

    <target name="jfxbe-profile-noscript" depends="-set-fallback-no-
javascript, jfxbe-profile"/>


</project>
```

CHAPTER 7

CODING AND TESTING

7.1 CODING

Once the design aspect of the system finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So, it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

7.3 TEST PROCEDURE

SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.4 TEST DATA AND OUTPUT

7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test

- Structure Test

7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

7.4.4 STRESS TEST

Stress Test is that test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

7.4.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

7.4.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.5 TESTING TECHNIQUES / TESTING STRATERGIES

7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise, the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

7.5.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

7.5.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods

- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.5.2 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer-based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

7.5.2.1 INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

7.5.2.2 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other errors in the program.

7.5.2.3 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- The function or performance characteristics confirm to specifications and are accepted.
- A validation from specification is uncovered and a deficiency created.
- Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

7.5.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

CHAPTER 8

CONCLUSION

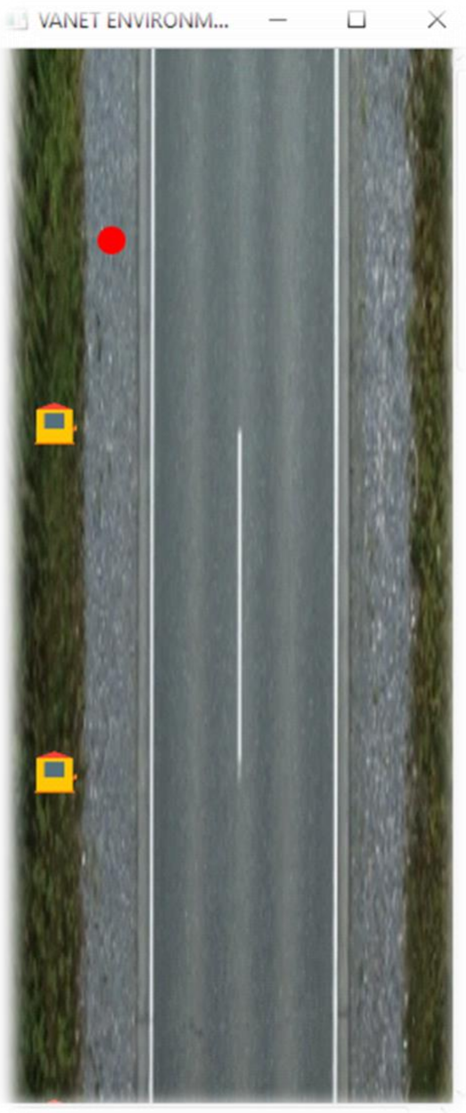
8.1 Conclusion and future enhancement:

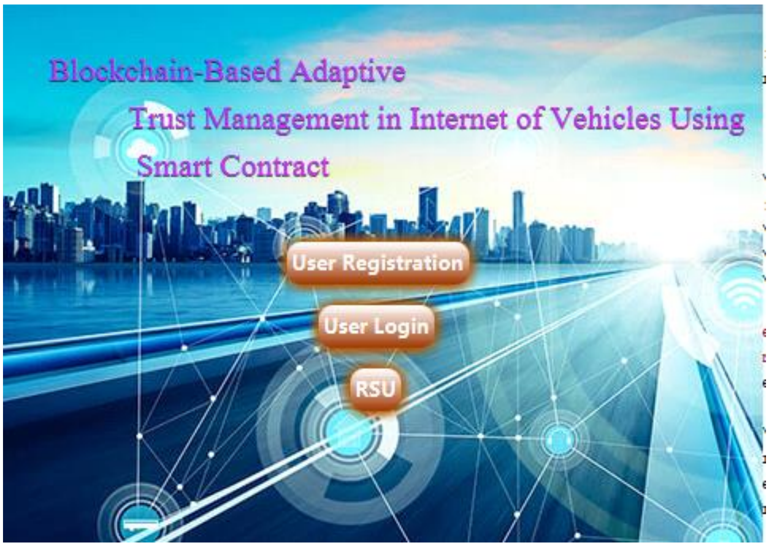
In this paper, we proposed mechanisms that manage trust using blockchain in IoV. We have provided a survey of existing works available in this increasingly important area. We proposed a blockchain-based decentralized approach in which CA/TA deployed the smart contract, and all RSUs work in a distributed manner to maintain consistent vehicular trust database and enhance reliability, availability, and consistency. We introduced the idea of maintaining sharded blockchains, that will not only reduce the propagation delay of transactions but will also increase the throughput and efficiency of the entire system. We also introduced incentive strategy for the vehicle participating in event detection, i.e., their contribution in the detection of a true event and its accurate reporting helps them to get rewards, which they can redeem for various services and payments. The proposed incentive mechanism encourages participating peers to perform well and get wallet points.

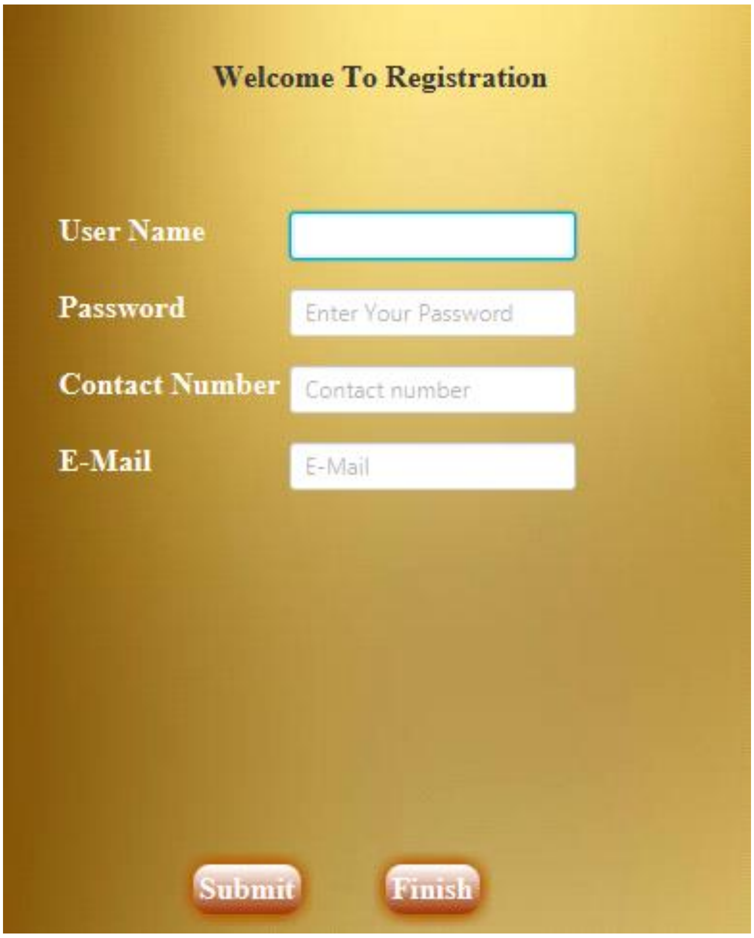
As future work, we will try to integrate the misbehaviour detection process and the privacy part. We will look for the role of AI in the misbehavior detection and efficient consensus algorithms in the RSU plane of IoV for decentralized trust management.

8.2 Appendices

A.1 Screenshots




Blockchain-Based Adaptive
Trust Management in Internet of Vehicles Using
Smart Contract
User Registration
User Login
RSU



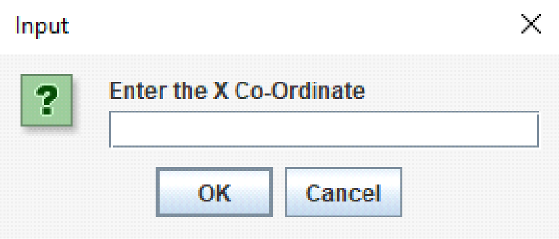
Welcome To Registration

User Name

Password: Enter Your Password

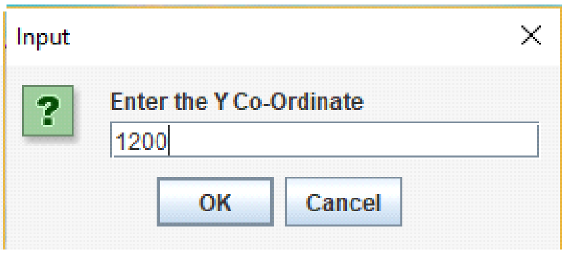
Contact Number: Contact number

E-Mail: E-Mail



Input

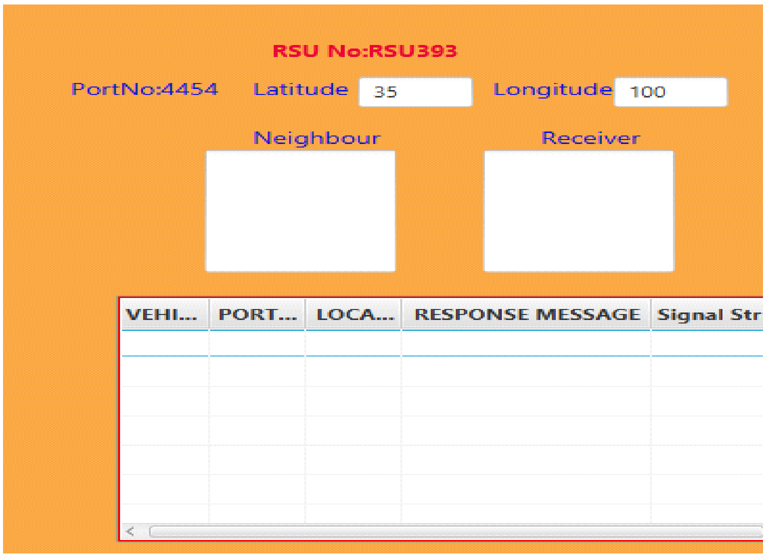
Enter the X Co-Ordinate



Input

Enter the Y Co-Ordinate

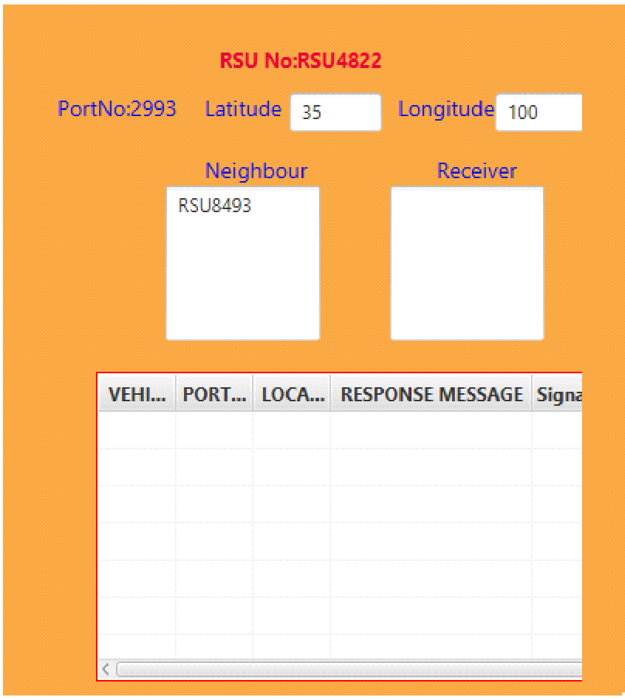
1200



RSU No:RSU393

PortNo:4454 Latitude 35 Longitude 100

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signal Str

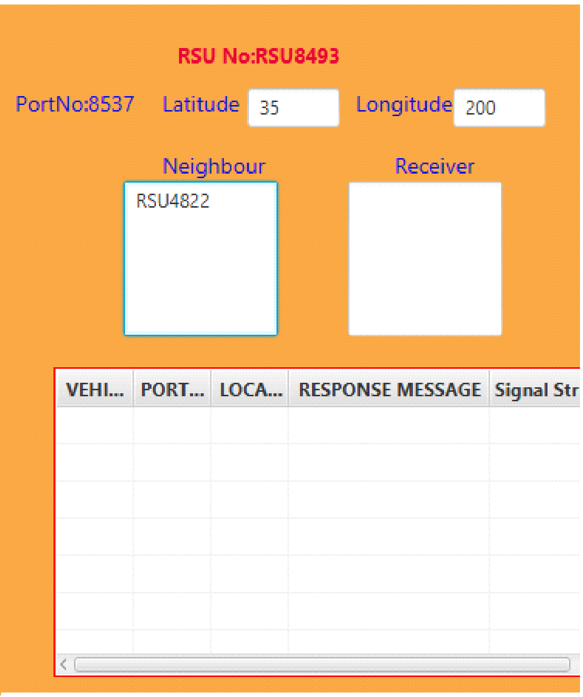


RSU No:RSU4822

PortNo:2993 Latitude 35 Longitude 100

Neighbour: RSU8493

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signa

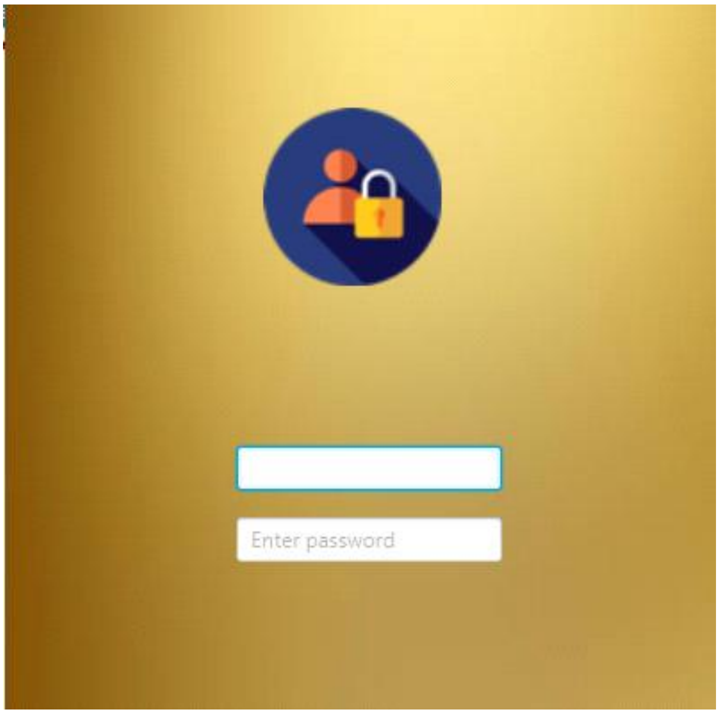


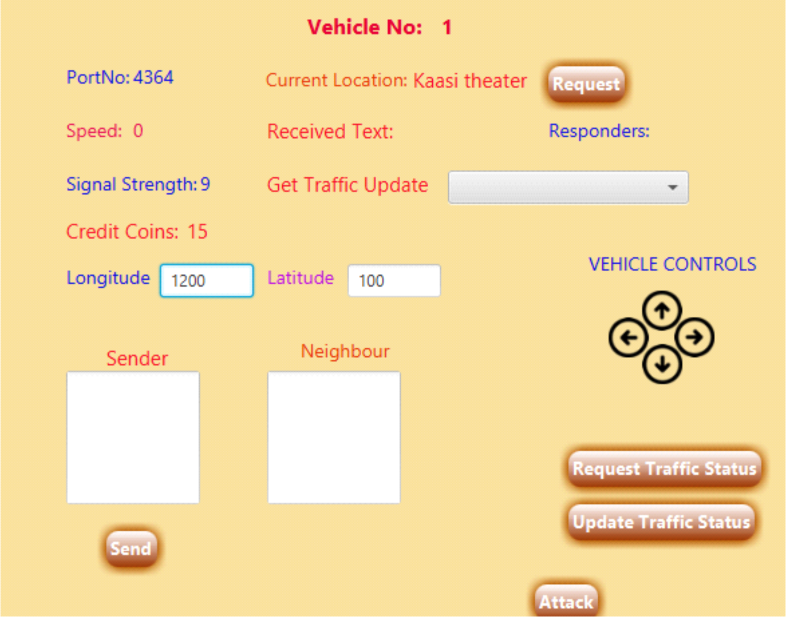
RSU No:RSU8493

PortNo:8537 Latitude 35 Longitude 200

Neighbour: RSU4822

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signal Str





Vehicle No: 1

PortNo: 4364

Current Location: Kaasi theater

Speed: 0

Received Text:

Responders:

Signal Strength: 9

Get Traffic Update

Credit Coins: 15

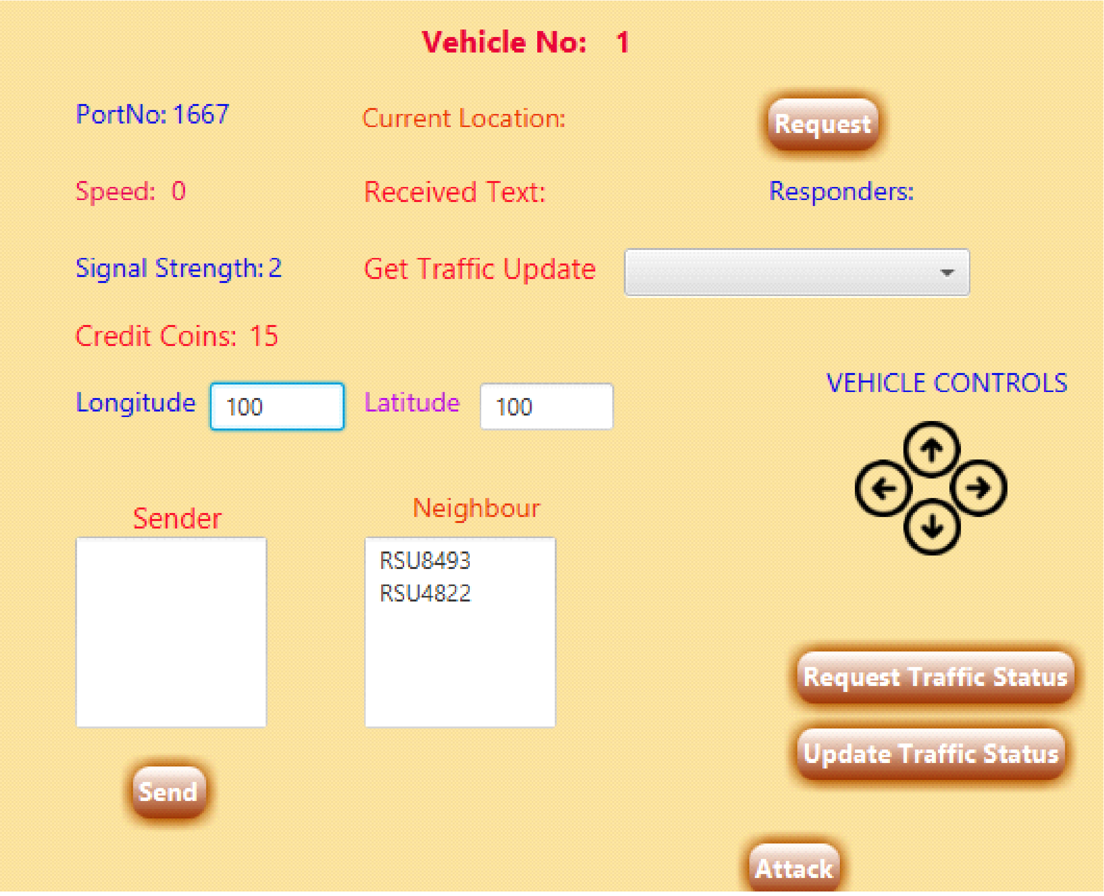
Longitude 1200

Latitude 100

VEHICLE CONTROLS

Sender

Neighbour



Vehicle No: 1

PortNo: 1667

Current Location:

Speed: 0

Received Text:

Responders:

Signal Strength: 2

Get Traffic Update

Credit Coins: 15

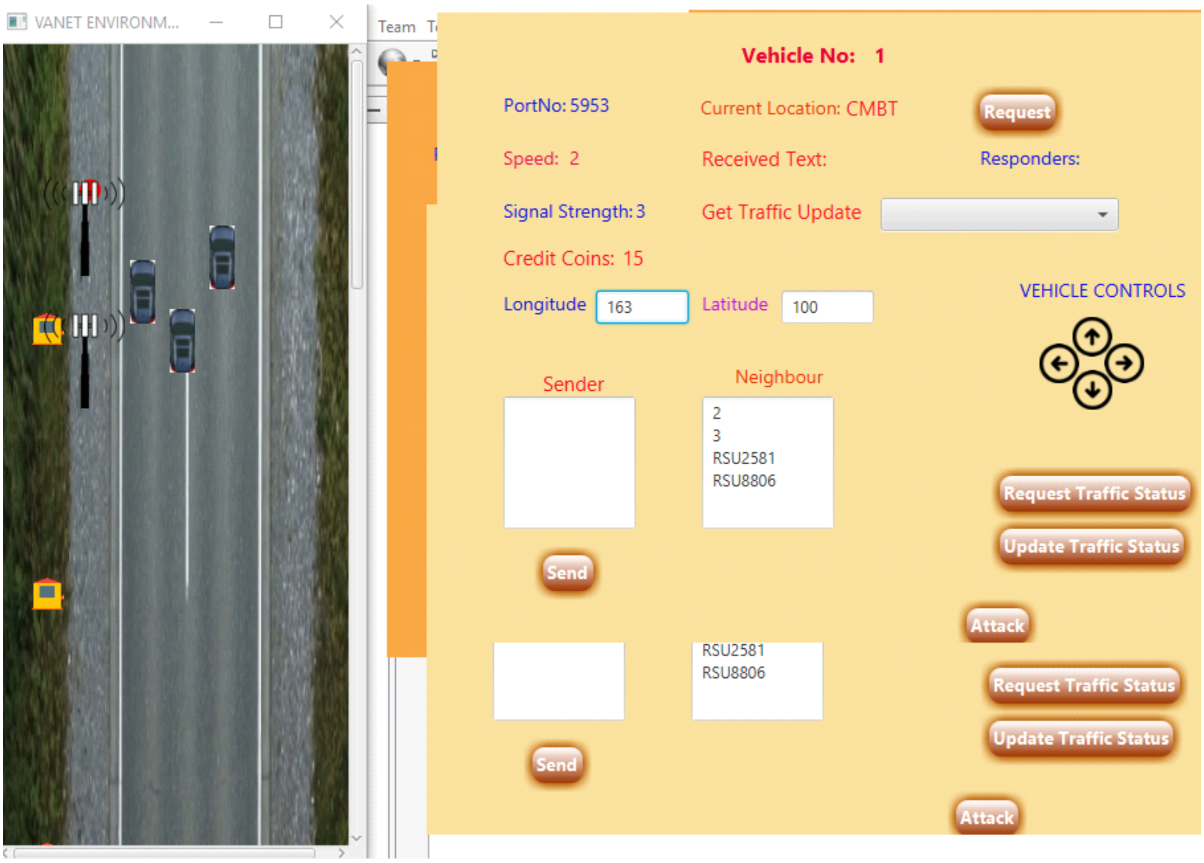
Longitude 100

Latitude 100

Sender

Neighbour

RSU8493
RSU4822



Vehicle No: 1

PortNo: 5953

Current Location: CMBT

Speed: 2

Received Text:

Responders:

Signal Strength: 3

Get Traffic Update

Credit Coins: 15

Longitude 163

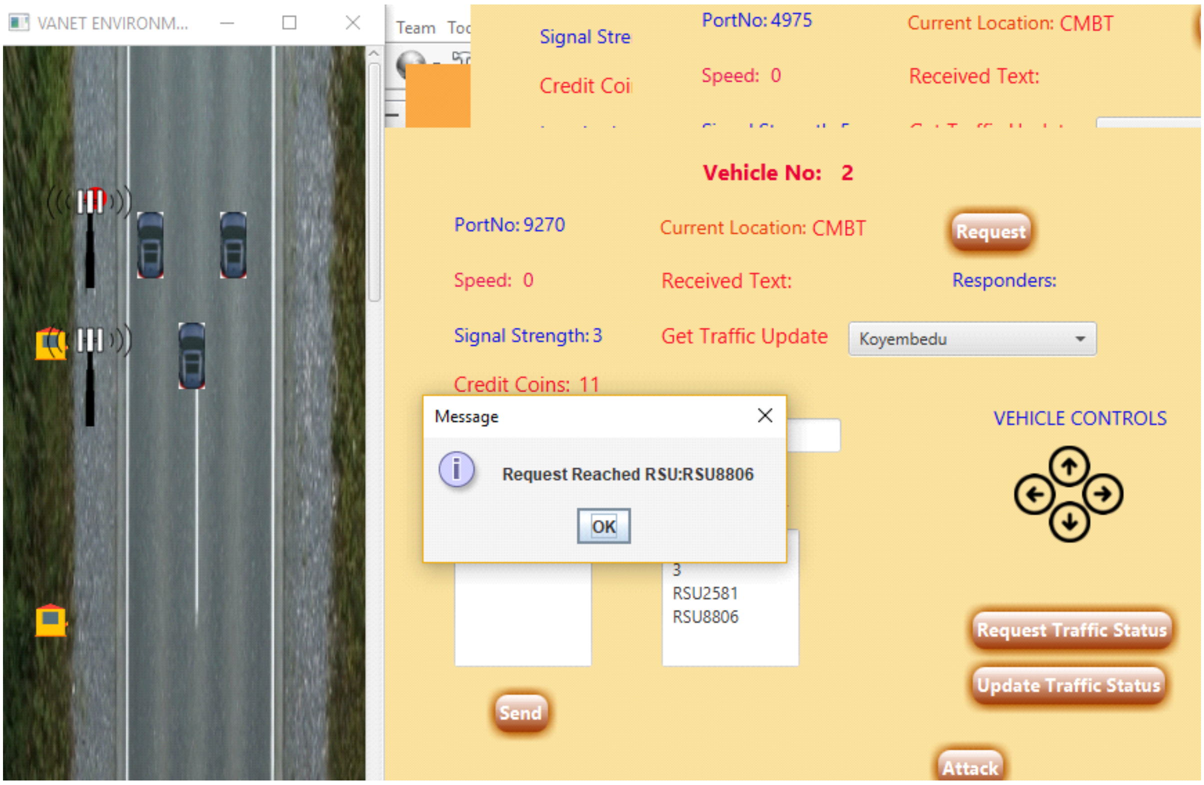
Latitude 100

Sender

Neighbour

2
3
RSU2581
RSU8806

RSU2581
RSU8806



Signal Stre

Credit Coi

PortNo: 4975

Speed: 0

Current Location: CMBT

Received Text:

Vehicle No: 2

PortNo: 9270

Current Location: CMBT

Speed: 0

Received Text:

Responders:

Signal Strength: 3

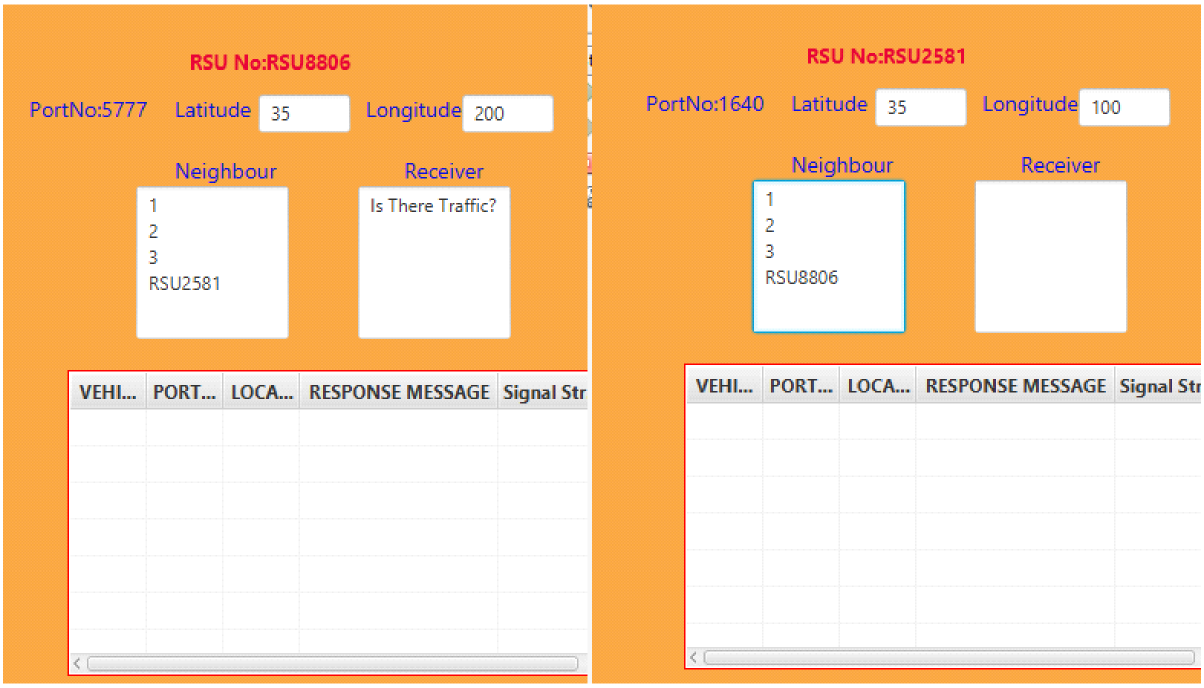
Get Traffic Update Koyembedu

Credit Coins: 11

Message

Request Reached RSU:RSU8806

3
RSU2581
RSU8806



RSU No:RSU8806

PortNo:5777 Latitude 35 Longitude 200

Neighbour: 1, 2, 3, RSU2581

Receiver: Is There Traffic?

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signal Str

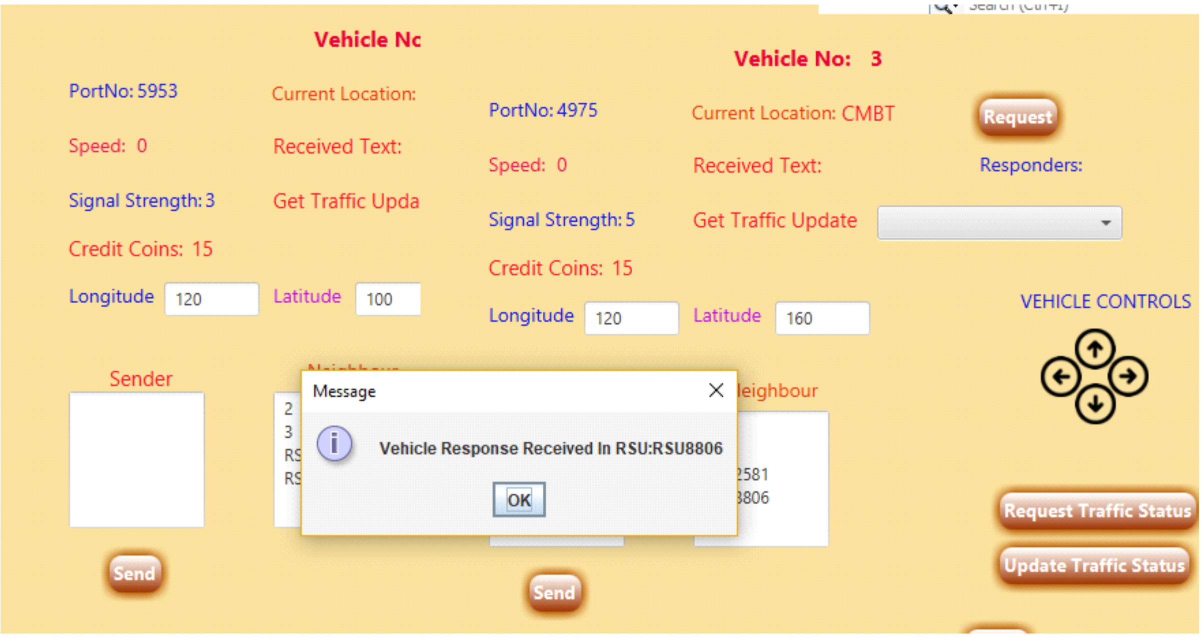
RSU No:RSU2581

PortNo:1640 Latitude 35 Longitude 100

Neighbour: 1, 2, 3, RSU8806

Receiver

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signal Str



Vehicle Nc

PortNo: 5953

Current Location:

Speed: 0

Received Text:

Signal Strength: 3

Get Traffic Upda

Credit Coins: 15

Longitude 120 Latitude 100

Vehicle No: 3

PortNo: 4975

Current Location: CMBT

Speed: 0

Received Text:

Responders:

Signal Strength: 5

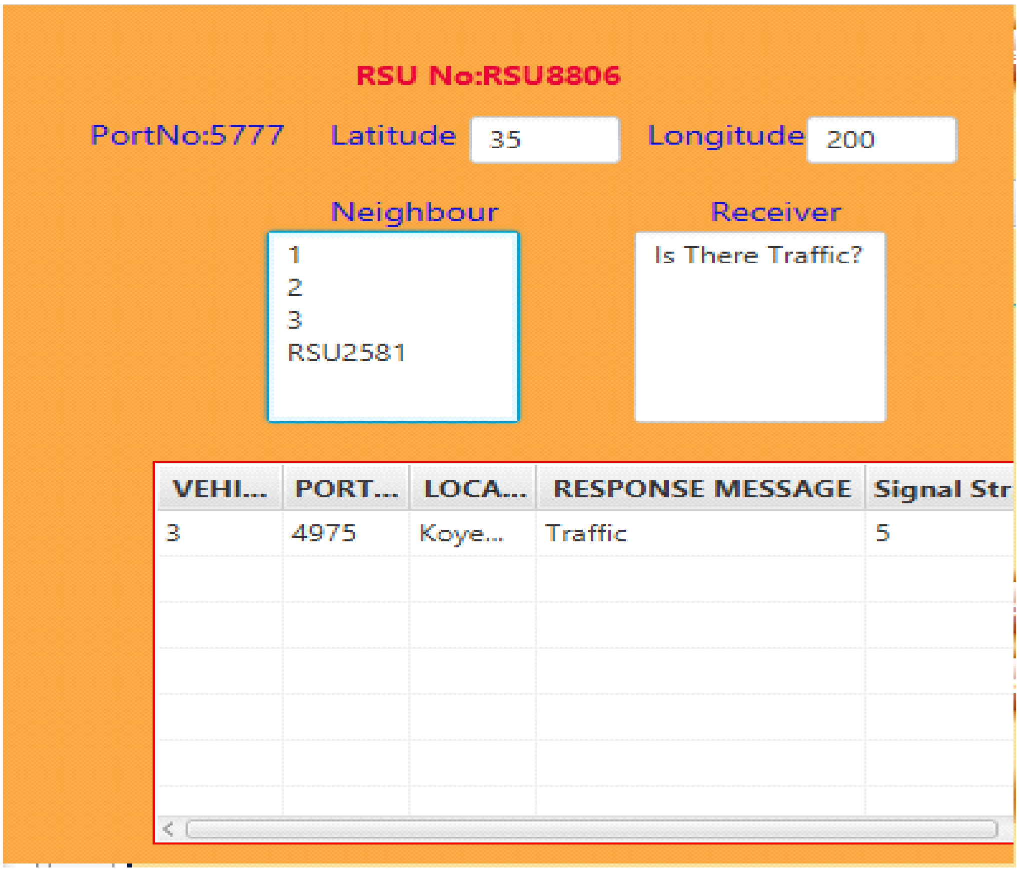
Get Traffic Update

Credit Coins: 15

Longitude 120 Latitude 160

Message

Vehicle Response Received In RSU:RSU8806



RSU No:RSU8806

PortNo:5777 Latitude 35 Longitude 200

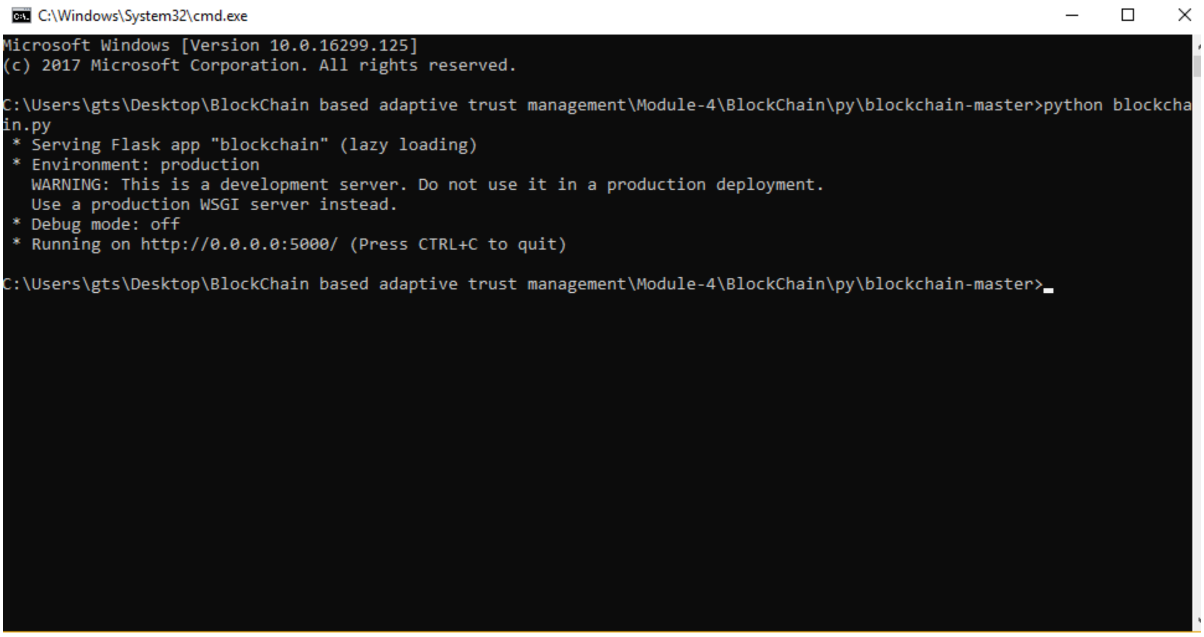
Neighbour

1
2
3
RSU2581

Receiver

Is There Traffic?

VEHI...	PORT...	LOCA...	RESPONSE MESSAGE	Signal Str
3	4975	Koye...	Traffic	5



```
Microsoft Windows [Version 10.0.16299.125]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\gts\Desktop\BlockChain based adaptive trust management\Module-4\BlockChain\py\blockchain-master>python blockcha
in.py
 * Serving Flask app "blockchain" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)

C:\Users\gts\Desktop\BlockChain based adaptive trust management\Module-4\BlockChain\py\blockchain-master>_
```

CHAPTER 9

REFERENCES

[1] J. Kamel, F. Haidar, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "A misbehavior authority system for Sybil attack detection in C-ITS," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON),* New York, NY, USA, 2019.

[2] P. K. Singh, S. K. Nandi, and S. Nandi, "A tutorial survey on vehicular communication state of the art, and future research directions," *Veh. Commun.*, vol. 18, Aug. 2019, Art. no. 100164.

[3] H. D. Abdulkarim and H. Sarhang, "Normalizing RSS values of Wi-Fi access points to improve an integrated indoors smartphone positioning solutions," in *Proc. Int. Eng. Conf. (IEC),* Jun. 2019, pp. 171–176.

[4] K. Z. Ghafoor, M. Guizani, L. Kong, H. S. Maghdid, and K. F. Jasim, "Enabling efficient coexistence of DSRC and C-V2X in vehicular networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 134–140, Apr. 2020.

[5] J. Huang, L. Kong, G. Chen, M.-Y. Wu, X. Liu, and P. Zeng, "Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.J. Huang *et al.*, "Blockchain based mobile crowd sensing in industrial systems," *IEEE Trans. Ind. Informat.*, published, Jan. 3, 2020,

[6] N. Torabi and B. S. Ghahfarokhi, "Survey of medium access control schemes for inter-vehicle communications," *Comput. Elect. Eng.*, vol. 64, pp. 450–472,

Nov. 2017

[7] R. Bauza and J. Gozálvez, “Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications,” *J. Netw. Comput. Appl.*, vol. 36, no. 5, pp. 1295–1307, 2013.

[8] A. Aliedani and S. W. Loke, “Cooperative car parking using vehicle-to-vehicle communication: An agent-based analysis,” *Comput. Environ. Urban Syst.*, Jun. 2018. [Online]. Available:https://doi.org/10.1016/j.compenvurbsys.2018.06.002

[9] A. Fernández-Isabel and R. Fuentes-Fernández, “Analysis of intelligent transportation systems using model-driven simulations,” *Sensors*, vol. 15, no. 6, pp. 14116–14141, 2015.

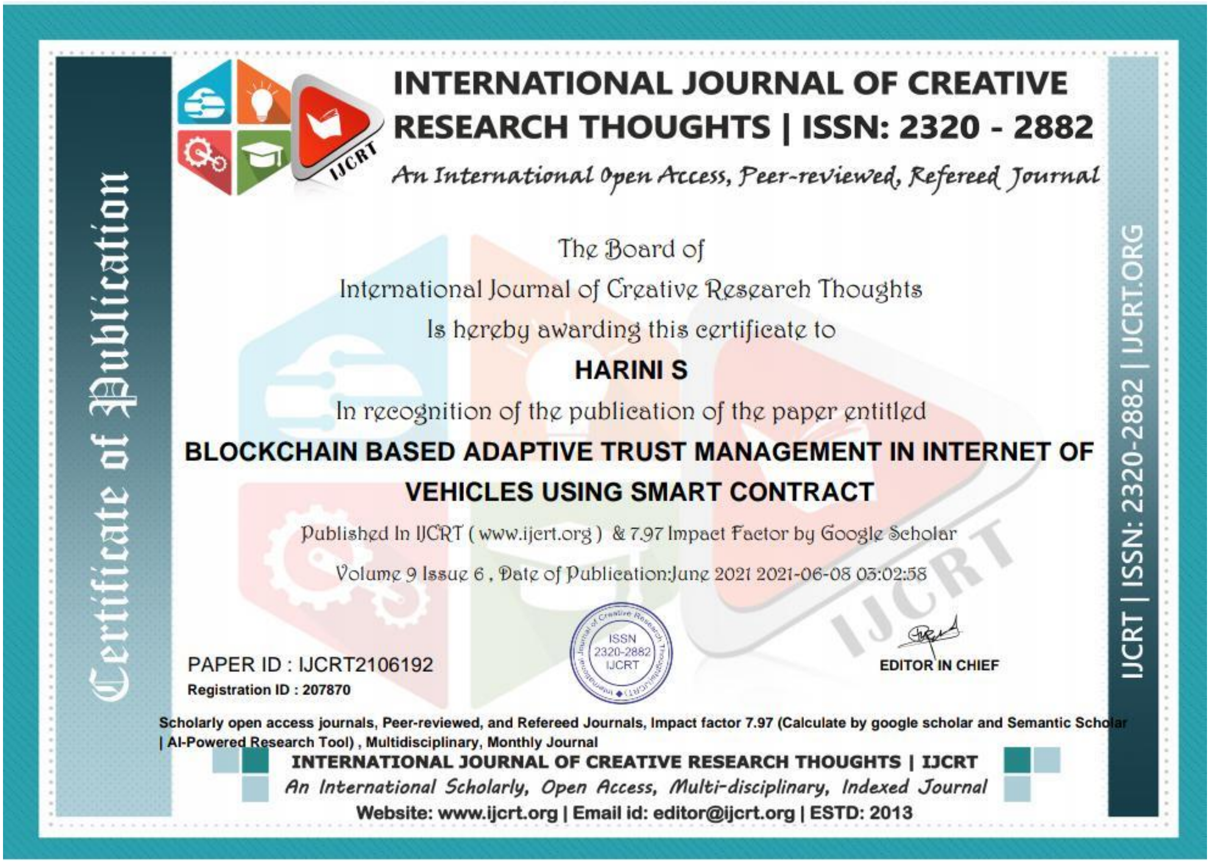
[10] S. Harrabi, I. B. Jaafar, and K. Ghedira, “A novel clustering algorithm based on agent technology for VANET,” *Netw. Protocols Algorithms*, vol. 8, no. 2, pp. 1–19, 2016.

[11] K. Grover, A. Lim, S. Lee, and Q. Yang, “Privacy-enabled probabilistic verification in broadcast authentication for vehicular networks,” *Ad Hoc Sensor Wireless Netw.*, vol. 32, nos. 3–4, pp. 239–274, 2016

CHAPTER 10

PUBLICATIONS

[1] HARINI S, AROCKIA SNEHA A, KAVIYA B, VIJAYALAKSHMI C, **"BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT"**, *International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.9, Issue 6, pp.b529-b532, June 2021*, Available at: http://www.ijcrt.org/papers/IJCRT2106192.pdf



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to

HARINI S

In recognition of the publication of the paper entitled

BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 6 , Date of Publication:June 2021 2021-06-08 03:02:58

PAPER ID : IJCRT2106192
Registration ID : 207870

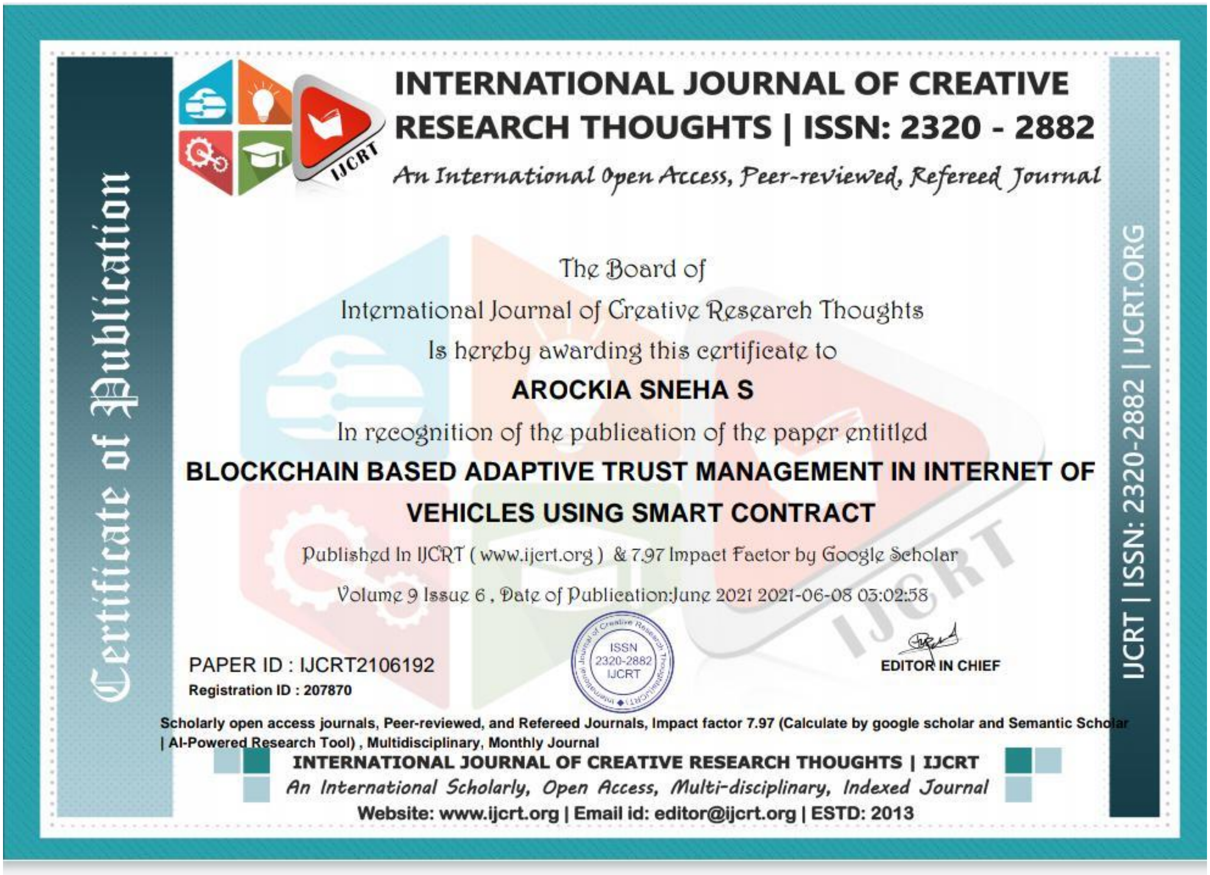
ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to

AROCKIA SNEHA S

In recognition of the publication of the paper entitled

BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 6 , Date of Publication:June 2021 2021-06-08 03:02:58

PAPER ID : IJCRT2106192
Registration ID : 207870

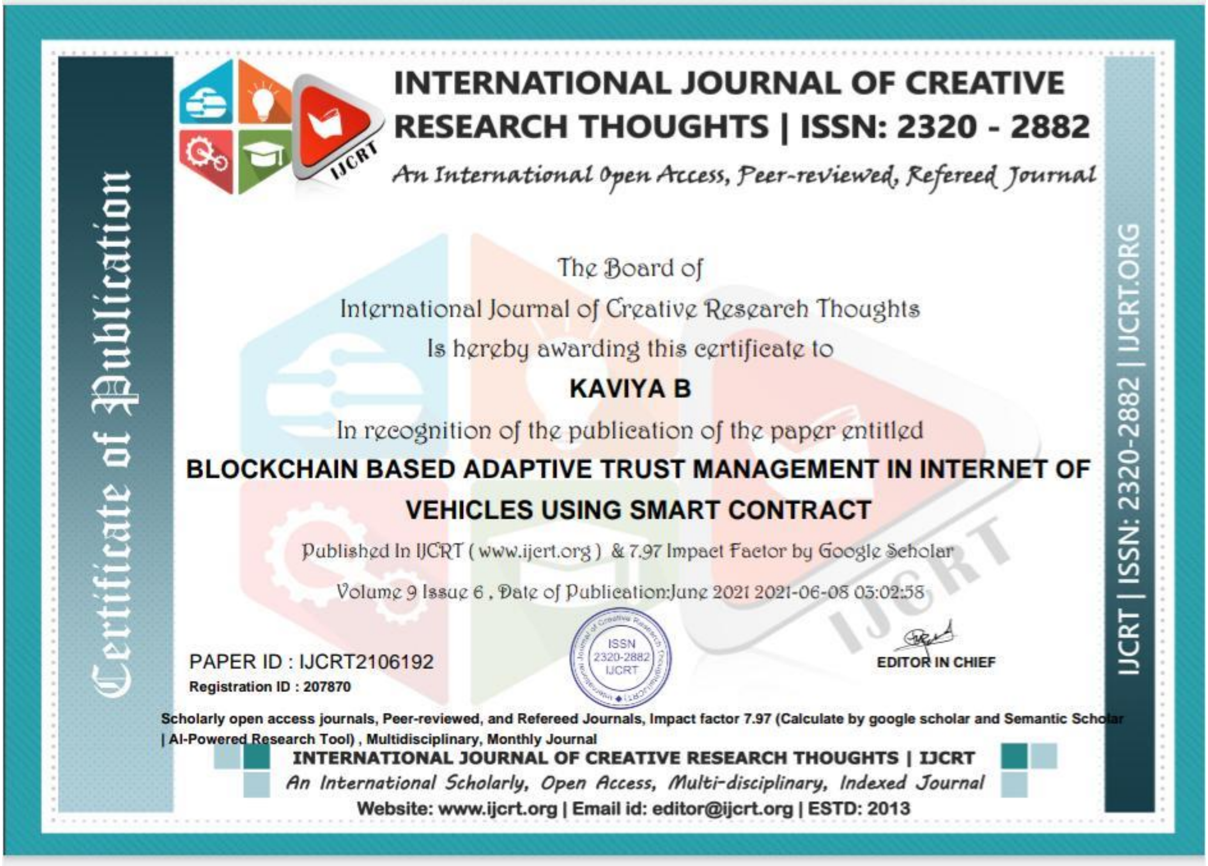
ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG



Certificate of Publication

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | ISSN: 2320 - 2882

An International Open Access, Peer-reviewed, Refereed Journal

The Board of
International Journal of Creative Research Thoughts
Is hereby awarding this certificate to

KAVIYA B

In recognition of the publication of the paper entitled

BLOCKCHAIN BASED ADAPTIVE TRUST MANAGEMENT IN INTERNET OF VEHICLES USING SMART CONTRACT

Published In IJCRT (www.ijcrt.org) & 7.97 Impact Factor by Google Scholar

Volume 9 Issue 6 , Date of Publication:June 2021 2021-06-08 03:02:58

PAPER ID : IJCRT2106192
Registration ID : 207870

ISSN 2320-2882 IJCRT

EDITOR IN CHIEF

Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS | IJCRT
An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal
Website: www.ijcrt.org | Email id: editor@ijcrt.org | ESTD: 2013

IJCRT | ISSN: 2320-2882 | IJCRT.ORG