

**MONITORING PATIENT HEALTH DATA
WITH MOBILE EDGE BASED IOT WITH
ONE TIME PAD ENCRYPTION**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

R. GOMATHI (38119007)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **R. GOMATHI (38119007)** who carried out the project entitled “**Monitoring patient health data with Mobile edge Based IOT with one time pad encryption**” under my supervision From November 2021 to April 2022

Internal Guide

Dr. L. K. Joshila Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI M.E., Ph. D/ Dr. L. LAKSHMANAN M.E., Ph. D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I R. GOMATHI (38119007) hereby declare that the Project Report entitled **“MONITORING PATIENT HEALTH DATA WITH MOBILE EDGE BASED IOT WITH ONE TIME PAD ENCRYPTION”** done by us under the guidance of Dr. L. K. Joshila is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in computer science

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing , Dr.S.Vigneshwari M.E., Ph.D. and Dr.L.Lakshmanan M.E., Ph.D. ,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. L. K. Joshila** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of

ABSTRACT

In the current trend Internet of Things (IoT) technology is more useful in healthcare in terms of mobile health and remote patient monitoring. IoT generates an unprecedented amount of data that can be processed using cloud computing. But for real-time remote health monitoring applications, the delay caused by transferring data to the cloud and back to the application is unacceptable. So when the health care IoT devices starts uploading current status of patient from smart home or hospital to the cloud continuously. The Fog-node collects the data and computes the received data and generates events for abnormal cases. When an event is triggered the fog-node sends alert to doctor, ambulance and relatives based on the threshold of event occurred. But in recent research installation of fog nodes and maintaining as a server is again a financial constraint to the organization, so we are going to implement the computation undergone in edge or fog servers in patient's mobile device with one-time-pad security. IoT based remote monitoring systems have been suggested by various researchers due to their high efficiency in delivering intensive time-sensitive information to the clients. In existing system the smart health IoT devices will keep observing the patients and send data to the cloud directly. This result in massive storage accumulation in cloud storage, processing the collected data is obviously a hard task to perform which results in computational overhead and delay in sending emergency alert. In Fog Layer edge level computing is performed based on the threshold values the health status of respective patient is analyzed. If the status is getting abnormal then the Fog layer sends the health data to the Cloud. Then the emergency message will be sent to Hospital Ambulance, Doctor and relatives based on the type of event triggered. Fog computing promotes highly efficient output because maximum computation is done edge level which makes the cloud process as light weight. Installation of fog nodes and maintaining as a server is again a financial constraint to the organization, so we going to implement the computation undergone in edge or fog servers in patient's mobile device with one-time-pad security.

TABLE OF CONTENTS

CHAPTER NUMBER	TITLE	PAGE NUMBER
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	INTRODUCTION	1
	1.1 AIM	1
	1.2 SYNOPSIS	1
	1.3 PROBLEM STATEMENT	1
	1.4 MOTIVATION	1
2	LITERATURE SURVEY	3
	LITERATURE SURVEY PAPERS	3
3	METHODOLOGY	8
	PROJECT PURPOSE AND SCOPE	8
	AIM AND OBJECTIVES	8
	PROGRAMMING LANGUAGES USED	9
	CONCEPTS AND METHODOLOGIES USED	19
	UML DIAGRAM	23
	MODULES	27
	PROJECT HARDWARE REQUIREMENTS	

CHAPTER NUMBER	TITLE	PAGE NO
4	RESULTS AND DISCUSSION	31
	4.1 WORKING	31
	4.2 RESULTS	42
5	SUMMARY AND CONCLUSION	44
	5.1 CONCLUSION	44
	REFERENCES	45
	APPENDIX	46

LIST OF FIGURES

FIG NO	FIGURE NAME	PAGE NO
3.1	JDK ARCHITECTURE	10
3.2	JRE ARCHITECTURE	11
3.3	JSP ARCHITECTURE	12
3.4	JSP LIFECYCLE	13
3.5	TOMCAT ARCHITECTURE	16
3.6	ANDROID ARCHITECTURE	19
3.7	IOT ARCHITECTURE	21
3.8	EDGE COMPUTING ARCHITECTURE	21
3.9	ONE TIME PAD	22
3.10	USE CASE DIAGRAM	23
3.11	LEVEL 0 DFD	23
3.12	LEVEL 1 DFD	24
3.13	LEVEL 2 DFD	24
3.14	LEVEL 3 DFD	25
3.15	ACTIVITY DIAGRAM	25
3.16	SEQUENCE DIAGRAM	26
3.17	COLLABORATION DIAGRAM	26
3.18	SYSTEM ARCHITECTURE	29
4.1	HOSPITAL WEB APPLICATION HOME PAGE	31
4.2	ADMIN SIGNUP PAGE	32
4.3	ADMIN CONTROL PAGE	32
4.4	DOCTOR LOGIN PAGE	33

FIG NO	FIGURE NAME	PAGE NO
4.5	DOCTOR PAGE AFTER LOGIN	34
4.6	PATIENT TAB	34
4.7	PATIENT SIGN UP TAB	35
4.8	PATIENTS LOGIN	36
4.9	PATIENTS MAKE APPOINTMENT PAGE	36
4.10	PROFILE PAGE	37
4.11	AMBULANCE PAGE	37
4.12	PATIENT'S MOBILE APP	38
4.13	MAKE APPOINTMENT THROUGH APP	39
4.14	KIT VALUE PROCESSING	39
4.15	DOCTOR LOGIN	40
4.16	PATIENT REPORT	41
4.17	AMBULANCE LOGIN	42

LIST OF ABBREVIATIONS

SI NO	ABBREVIATIONS	EXPANSION
1	JDK	Java development kit
2	JVM	Java Virtual Machine
3	JRE	Java Runtime Environment
4	JSP	Java server page
5	IOT	Internet of Things
6	OTP	One Time Pad

CHAPTER 1

INTRODUCTION

AIM

The main aim of our project is to perform event-based triggering to process the patient's real-time data and compute the data in patient's mobile and then to uploaded to cloud and also to secure the data with encryption.

SYNOPSIS

In the current trend Internet of Things (IoT) technology is more useful in healthcare in terms of mobile health and remote patient monitoring. IoT generates an unprecedented amount of data that can be processed using cloud computing. But for real-time remote health monitoring applications, the delay caused by transferring data to the cloud and back to the application is unacceptable. So, when the health care IoT devices starts uploading current status of patient from smart home or hospital to the cloud continuously. The patient's mobile will be acting as Fog-node in which it collects the data and computes the received data and generates events for abnormal cases. When an event is triggered the cloud sends alert to doctor, ambulance and relatives based on the threshold of event occurred. All data transmission occurring in-between Cloud Server and Patient' Mobile is encrypted using One Time Pad security mechanism.

PROBLEM STATEMENT

Due to continuous monitoring of IOT devices even the Fog Servers got accumulated with bulk of junk files. Processing the data in Fog Servers takes very long time which makes slow alerts and fog Servers are complicated to create and manage. To overcome the disadvantages of fog we create a project based on Secured OTP IOT Environment in Medical Field. Here we use Mobile edge computing concept to make computations which require less resources and is faster and less complicated than fog. We create a mobile edge-based health monitoring system which monitors patient's health continuously and automatically computes the status of patient and sends emergency SMS to particular members based on the status of patient.

MOTIVATION

There are large cases of sudden deaths happening all around the world due to heart attack , breathing trouble and carelessness especially for old people This project is mainly dedicated to reduce such sudden unexpected deaths and save lives .Family members fear for Heart patients in their family ,So To reduce the fear and provide 24/7 monitoring service This project is dedicated .Doctor fees are high nowadays in private hospitals .To overcome this and Allow users Reduce doctor expenses and to automate the process of Calculating various health factors we dedicate this project.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY PAPERS

- [1] Paper Title:** Comparative performance and energy consumption analysis of different AES implementations on a Wireless Sensor Network Node

Author Name: Fan Zhang, Reiner Dojen, Tom Coffey

Paper Overview:

Many of the diverse Wireless Sensor Network (WSN) applications, such as home automation, traffic control and medical patient monitoring, have a strong requirement for the confidentiality of sensing data. At the same time, the restricted resources of WSN nodes demand that applications are implemented as efficiently as possible. Recently, the Advanced Encryption Standard (AES) has been proposed as the preferable symmetric cipher for WSN applications. This paper presents a performance and energy consumption analysis of three AES implementations on a wireless sensor node. For each implementation the memory requirements, the execution time and the energy consumption is monitored. The results of the presented analysis show that while the hardware supported AES is faster than the software implementations it is less energy efficient, due to increased power requirements of the additional hardware.

Pros: The restricted resources of WSN nodes demand that applications are implemented as efficiently as possible. The execution time and the energy consumption is monitored.

Cons: .Ad hoc networks composed of small network nodes with low power capacities, small resource storages and relatively low computation capabilities..The decrease of physical node sizes and the increase of nodes in WSN systems makes the replacement of depleted batteries more and more impractical.

- [2] Paper Title:** LFSR-based Hashing and Authentication

Author Name: Hugo Krawceyk

Paper overview:

We present simple and efficient hash functions applicable to secure authentication of information. The constructions are mainly intended for message authentication in systems implementing stream cipher encryption and are suitable for other applications as well. The proposed hash functions are implemented through linear feedback shift registers and therefore

attractive for hardware applications. **A5** an example, a single **64** bit LFSR will be used to authenticate 1 Gbit of information with a failure probability of less than 2^{-32} . One of the constructions is the cryptographic version of the well known cyclic redundancy codes (CRC); the other is based on Toeplitz hashing where the matrix entries are generated by a **LFSR**. The later construction achieves essentially the same hashing and authentication strength of a completely random matrix but at a substantially lower cost in randomness, key size and implementation complexity. Of independent interest is **our** characterization of the properties required from a family of hash functions in order to be secure for authentication when combined with a (secure) stream cipher.

Pros: Hash functions are implemented through linear feedback shift registers and therefore attractive for hardware applications. 2. Constructions are mainly intended for message authentication in systems implementing stream cipher encryption and are suitable for other applications as well

Cons: The later construction achieves essentially the same hashing and authentication. Key size and implementation complexity

[3] Paper Title: Frank Miller: Inventor of the One-Time Pad

Author Name: STEVEN M. BELLOVIN

Paper overview:

The invention of the one-time pad is generally credited to Gilbert S. Vernam and Joseph O. Mauborgne. We show that it was invented about 35 years earlier by a Sacramento banker named Frank Miller. We identify which Frank Miller it was, and speculate on what might have led him to his idea. We also discuss whether or not Mauborgne might have known of Miller's work, especially via his colleague Parker Hitt.

Pros : One-time pads are in theory the strongest possible algorithmic cipher: if the key is used properly, they cannot be broken, even in theory. It would have had the same property of absolute security. The telegraph code is conventional for its time: it mapped phrases into both English words and 5-digit code groups.

Cons: In theory, Miller can claim priority, reality is more complex. It does not appear to be a particularly good compression system even when ranked against its peers.

[4] Paper Title: Securing Fog Computing for Internet of Things Applications: Challenges and Solutions

Author Name: Jianbing Ni, Kuan Zhang, Xiaodong Lin, Xuemin (Sherman) Shen,

Paper overview:

Internet of Things (IoT) allows billions of physical objects to be connected to collect and exchange data for offering various applications, such as environmental monitoring, infrastructure management and home automation. On the other hand, IoT has unsupported features (e.g., low latency, location awareness and geographic distribution) that are critical for some IoT applications, including smart traffic lights, home energy management and augmented reality. To support these features, fog computing is integrated into IoT to extend computing, storage and networking resources to the network edge. Unfortunately, it is confronted with various security and privacy risks, which raise serious concerns towards users. In this survey, we review the architecture and features of fog computing and study critical roles of fog nodes, including real-time services, transient storage, and data dissemination and decentralized computation. We also examine fog-assisted IoT applications based on different roles of fog nodes. Then, we present security and privacy threats towards IoT applications and discuss the security and privacy requirements in fog computing. Further, we demonstrate potential challenges to secure fog computing and review the state-of-the-art solutions used to address security and privacy issues in fog computing for IoT applications. Finally, by defining several open research issues, it is expected to draw more attention and efforts into this new architecture.

Pros: Fog computing is integrated into IoT to extend computing, storage and networking resources to the network edge. The growth of IoT leads to the generation of large amounts of data, which possess massive computing resources, storage space and communication bandwidth

Cons: IoT has unsupported features such as low latency, location awareness and geographic distribution. These are critical for some IoT applications, including smart traffic lights, home energy management and augmented reality.

[5] Project Title: Mini Sec: A Secure Sensor Network Communication Architecture

Author Name: Mark Luk, Ghita Mezzour, Adrian Perrig, Virgil Gligor

Paper overview:

Secure sensor network communication protocols need to provide three basic properties: data secrecy, authentication, and replay protection. Secure sensor network link layer protocols such as Tiny- Sec and ZigBee enjoy significant attention in the community. However, TinySec

achieves low energy consumption by reducing the level of security provided. In contrast, Zig Bee enjoys high security, but suffers from high energy consumption. Mini Sec is a secure network layer that obtains the best of both worlds: low energy consumption and high security. Mini Sec has two operating modes, one tailored for single-source communication, and another tailored for multi-source broadcast communication. The latter does not require per-sender state for replay protection and thus scales to large networks. We present a publicly available implementation of MiniSec for the Telos platform, and experimental results demonstrate our low energy utilization.

Pros: MiniSec is a secure network layer that obtains the best of both worlds: low energy consumption and high security. It does not require per-sender state for replay protection and thus scales to large networks.

Cons: Mini-sec is desirable in this setting because of the stringent energy constraints of sensor networks. It would be too expensive to run the counter resynchronization protocol among many receivers.

- [6] **Project Title:** Edge Computing for the Internet of Things: A Case Study
Author Name: Gopika PremSankar, Mario Di Francesco, and Tarik Taleb

Paper Overview

The amount of data generated by sensors, actuators and other devices in the Internet of Things (IoT) has substantially increased in the last few years. IoT data are currently processed in the cloud, mostly through computing resources located in distant data centers. As a consequence, network bandwidth and communication latency become serious bottlenecks. This article advocates edge computing for emerging IoT applications that leverage sensor streams to augment interactive applications. First, we classify and survey current edge computing architectures and platforms, then describe key IoT application scenarios that benefit from edge computing. Second, we carry out an experimental evaluation of edge computing and its enabling

Pros: Network-intensive data can be processed and analyzed just one hop away from end-devices, thereby reducing the bandwidth demands on network links to distant data centers. Edge computing servers also provide computational capabilities and can host interactive applications that support user mobility.

Cons: The increasing amount of data exchanged adds substantial stress on the network links to the cloud. The edge of the network is not clearly defined and the nodes expected to participate at the edge can vary.

CHAPTER 3

METHODOLOGY

PROJECT PURPOSE AND SCOPE

PURPOSE

The main aim of our project is to perform event-based triggering to process the patient's real-time data and compute the data in patient's mobile and then data got uploaded to cloud.

PROJECT SCOPE

We propose the Secured OTP IOT Environment in Medical Field. Here the patient's android mobile device will be acting as the edge server for doing computing with the receiving data from health monitoring devices. The health-related data get uploaded into Mobile node. In Mobile Edge Layer, edge level computing is performed based on the threshold values the health status of respective patient is analyzed. If the status is getting abnormal then the Mobile Edge layer sends the health data with One-time pad security to the Cloud. Then the emergency message will be sent to Hospital Ambulance, Doctor and relatives based on the type of event triggered. Mobile computing promotes highly efficient output because maximum computation is done edge level which makes the cloud process as light weight. We going to implement the computation in patient's mobile device with one-time-pad security.

AIM AND OBJECTIVES

- The main objective of our project is to perform event-based triggering to process the patient's real-time data at Fog Layer and then upload the data to cloud.
- To continuously monitor the patient's health status.
- To send alert messages to known members of the patient
- To Provide security to the user's medical data so unauthorized members cannot access the data
- To Automate the process of arranging the ambulance in case of critical stage.

- To make the user care about his/her health and create awareness about being healthy and fit.
- Make user understand his/her health conditions in most understandable way

PROGRAMMING LANGUAGES USED

The programming concepts used for developing the system are

1. J2SE-Core java
2. JSP-Java server page for developing web server
3. J2ME-Java for mobile application development.

CORE JAVA

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.^[16] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them. As of 2019, Java was one of the most popular programming languages in use according to GitHub particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

TOOLS REQUIRED FOR RUNNING JAVA

The following tools are required for the system to support java programming and successfully compile and execute java programs

- JDK-Java development Kit
- JRE-Java runtime environment

JDK-Java development kit

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development. Java developers are initially presented with two JDK tools, java and javac. Both are run from the command prompt. Java source files are simple text files saved with an extension of .java. After writing and saving Java source code, the javac compiler is invoked to create .class files. Once the .class files are created, the 'java' command can be used to run the java program. For developers who wish to work in an integrated development environment (IDE), a JDK bundled with Netbeans can be downloaded from the Oracle website. Such IDEs speed up the development process by introducing point-and-click and drag-and-drop features for creating an application. There are different JDKs for various platforms. The supported platforms include Windows, Linux and Solaris. Mac users need a different software development kit, which includes adaptations of some tools found in the JDK.

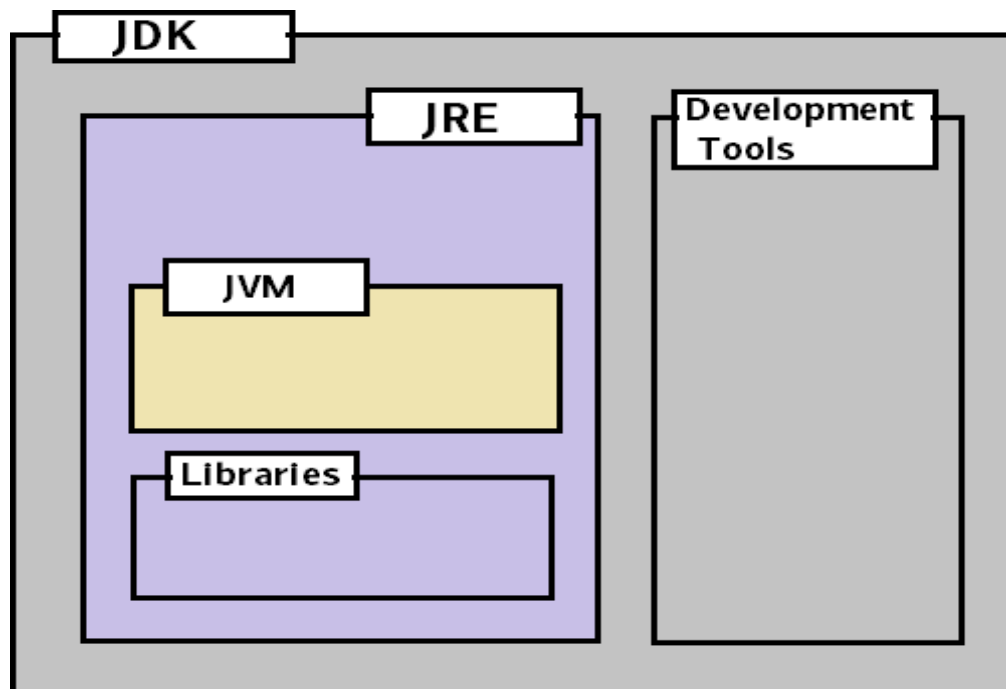


Figure 3.1-JDK architecture

JRE-JAVA RUNTIME ENVIRONMENT

The Java Runtime Environment (JRE) is a set of software tools for development of Java applications. It combines the Java Virtual Machine (JVM), platform core classes and supporting libraries is part of the Java Development Kit (JDK), but can be downloaded separately. JRE was originally developed by Sun Microsystems Inc., a wholly-owned subsidiary of Oracle Corporation. Also known as Java runtime. JRE consists of the following components: Deployment technologies, including deployment, Java Web Start and Java Plug-in. User interface toolkits, including Abstract Window Toolkit (AWT), Swing, Java 2D, Accessibility, Image I/O, Print Service, Sound, drag and drop (DnD) and input methods. Integration libraries, including Interface Definition Language (IDL), Java Database Connectivity (JDBC), Java Naming and Directory Interface (JNDI), Remote Method Invocation (RMI), Remote Method Invocation Over Internet Inter-Orb Protocol (RMI-IIOP) and scripting. Other base libraries, including international support, input/output (I/O), extension mechanism, Beans, Java Management Extensions (JMX), Java Native Interface (JNI), Math, Networking, Override Mechanism, Security, Serialization and Java for XML Processing (XML JAXP). Lang and util base libraries, including lang and util, management, versioning, zip, instrument, reflection, Collections, Concurrency Utilities, Java Archive (JAR), Logging, Preferences API, Ref Objects and Regular Expressions. Java Virtual Machine (JVM), including Java HotSpot Client and Server Virtual Machines 1.0 has evolved with a variety of class and package additions to the core libraries, and it has grown from a few hundred classes to several thousand in Java 2 Platform, Standard Edition (J2SE). Entirely new APIs have been introduced, and many of the original version 1.0 APIs have been deprecated. There has been a significant degree of criticism because of these massive additions.

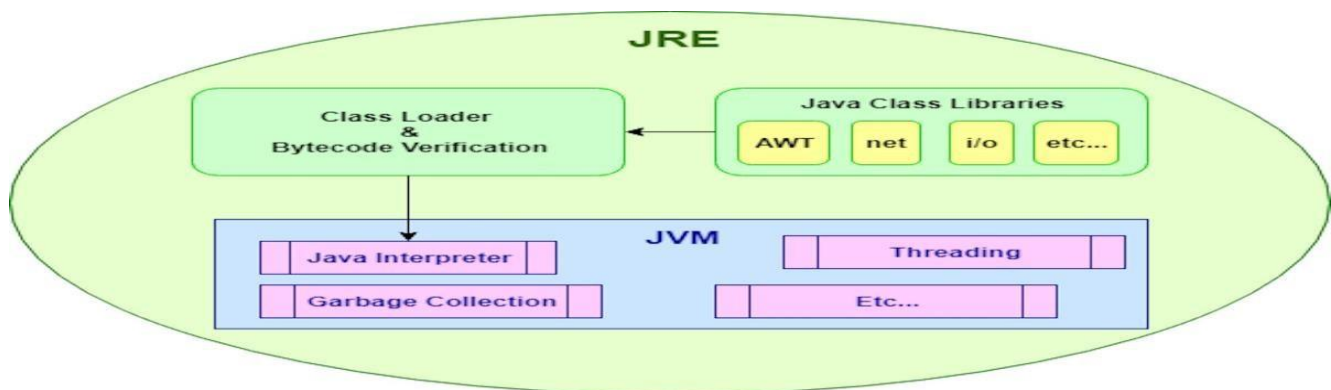


Figure 3.2 – JRE architecture

JAVA SERVER PAGE

It stands for Java Server Pages. It is a server-side technology. It is used for creating web application. It is used to create dynamic web content. In this JSP tags are used to insert JAVA code into HTML pages. It is an advanced version of Servlet Technology. It is a Web based technology helps us to create dynamic and platform independent webpages. In this, Java code can be inserted in HTML/ XML pages or both .JSP is first converted into servlet by JSP container before processing the client's request .JSP pages are more advantageous than Servlet:They are easy to maintain.No recompilation or redeployment is required.JSP has access to entire API of JAVA .JSP are extended version of Servlet.

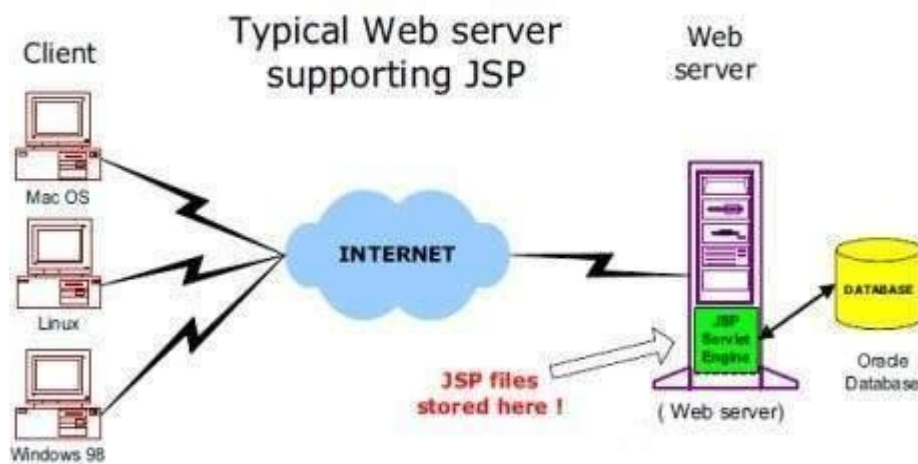


Figure 3.3 -JSP Architecture

JSP LIFE CYCLE

JSP Life Cycle is defined as translation of JSP Page into servlet as a JSP Page needs to be converted into servlet first in order to process the service requests. The Life Cycle starts with the creation of JSP and ends with the disintegration of that. When the browser asks for a JSP, JSP engine first checks whether it needs to compile the page. If the JSP is last compiled or the recent modification is done in JSP, then the JSP engine compiles the page.

Compilation process of JSP page involves three steps:

- Parsing of JSP
- Turning JSP into servlet
- Compiling the servlet

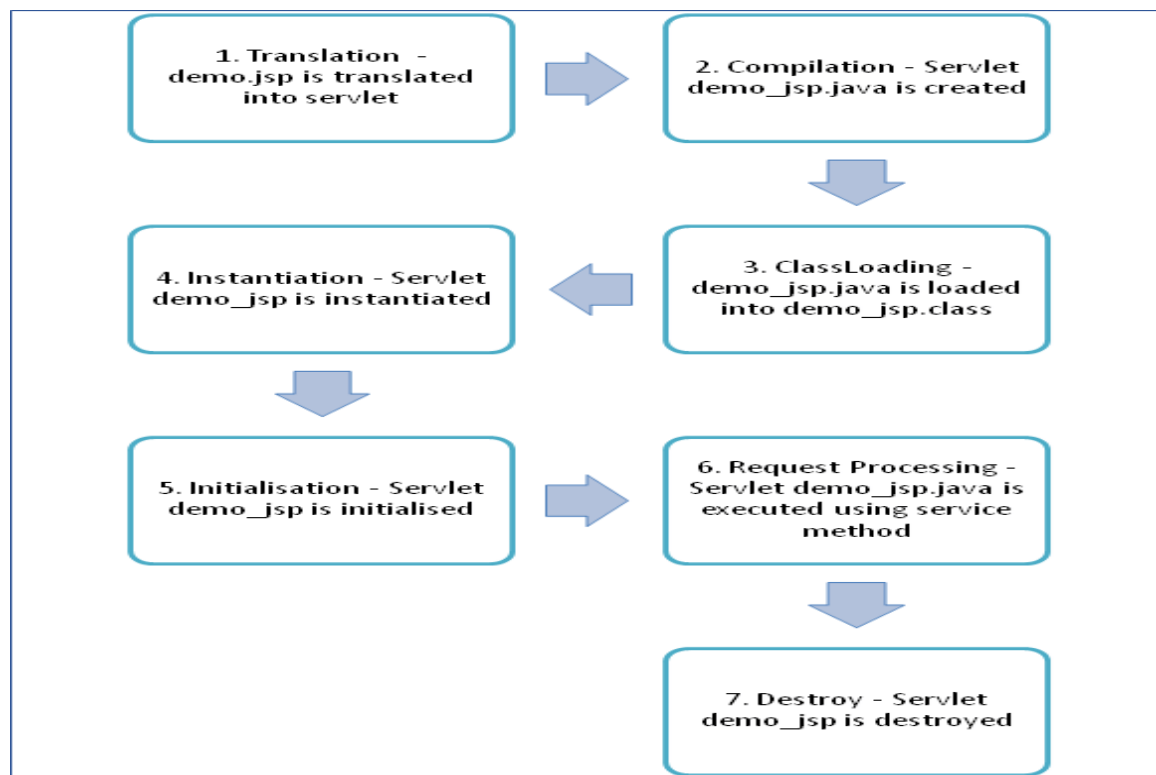


Figure 3.4-JSP Life cycle

Following steps explain the JSP life cycle:

1. Translation of JSP page
2. Compilation of JSP page(Compilation of JSP page into _jsp.java)
3. Classloading (_jsp.java is converted to class file _jsp.class)
4. Instantiation(Object of generated servlet is created)
5. Initialisation(_jspinit() method is invoked by container)
6. Request Processing(_jspservice() method is invoked by the container)
7. Destroy (_jspDestroy() method invoked by the container)

Translation of the JSP Page:

A Java servlet file is generated from a JSP source file. This is the first step of JSP life cycle. In translation phase, container validates the syntactic correctness of JSP page and tag files. The JSP container interprets the standard directives and actions, and the custom actions referencing tag libraries (they are all part of JSP page and will be discussed in the later section) used in this JSP page.

Compilation of the JSP Page

The generated java servlet file is compiled into java servlet class. The translation of java source page to its implementation class can happen at any time between the deployment of JSP page into the container and processing of the JSP page. In the above pictorial description demo_jsp.java is compiled to a class file demo_jsp.class

Classloading

- Servlet class that has been loaded from JSP source is now loaded into the container

Instantiation

In this step the object i.e. the instance of the class is generated. The container manages one or more instances of this class in the response to requests and other events. Typically, a JSP container is built using a servlet container. A JSP container is an extension of servlet container as both the container support JSP and servlet. A JSPPage interface which is provided by container provides init() and destroy () methods. There is an interface HttpJSPPage which serves HTTP requests, and it also contains the service method.

Initialization

```
public void jspInit()
{
    //initializing the code
}
```

jspinit() method will initiate the servlet instance which was generated from JSP and will be invoked by the container in this phase. Once the instance gets created, init method will be invoked immediately after that. It is only called once during a JSP life cycle, the method for initialization is declared as shown above

Request processing

```
void _jspService(HttpServletRequest request HttpServletResponse response)
{
    //handling all request and responses
```

```
}
```

jspervice() method is invoked by the container for all the requests raised by the JSP page during its life cycle. For this phase, it has to go through all the above phases and then only service method can be invoked. It passes request and response objects. This method cannot be overridden. The method is shown above: It is responsible for generating of all HTTP methods i.e. GET, POST, etc.

Destroy

```
public void _jspdestroy()
{
    //all clean up code
}
```

jspdestroy() method is also invoked by the container. This method is called when container decides it no longer needs the servlet instance to service requests. When the call to destroy method is made then, the servlet is ready for a garbage collection. This is the end of the life cycle. We can override jspdestroy() method when we perform any cleanup such as releasing database connections or closing open files.

TOOLS REQUIRED TO RUN JSP

The following tools are required to run jsp code easily without any error.

- Apache tomcat server
- Netbeans IDE for java

APACHE TOMCAT SERVER

Apache Tomcat is a long-lived, open source Java servlet container that implements several core Java enterprise specs, namely the Java Servlet, JavaServer Pages (JSP), and WebSocket's APIs. An Apache Software Foundation project, Tomcat was first released in 1998, just four years after Java itself. Tomcat started as a reference implementation for the first Java Servlet API and the JSP spec. While it's no longer the reference implementation for either of

these technologies, Tomcat remains the most widely used Java server, boasting a well-tested and proven core engine with good extensibility. In this short introduction, you'll learn why many software shops choose Tomcat for running Java web apps. You'll get an overview of Tomcat and how it's used, as well as installation instructions for the most current version as of this writing. The Java ecosystem supports several kinds of application server, so let's disambiguate them and see where Tomcat fits in: A servlet container is an implementation of the Java Servlet specification, used primarily for hosting Java servlets. A web server is a server designed to serve files from the local system, like Apache. A Java enterprise application server is a full-blown implementation of the Java EE (now Jakarta EE) specification. At heart, Tomcat is a servlet and JSP container. A Java servlet encapsulates code and business logic and defines how requests and responses should be handled in a Java server. JSP is a server-side view rendering technology. As the developer, you write the servlet or JSP page, then let Tomcat handle the routing. Tomcat also contains the Coyote engine, which is a web server. Thanks to Coyote, it's possible to extend Tomcat to include a variety of Java enterprise specs and capabilities, including the Java Persistence API (JPA). Tomcat also has an extended version, called TomEE, that includes more enterprise features.

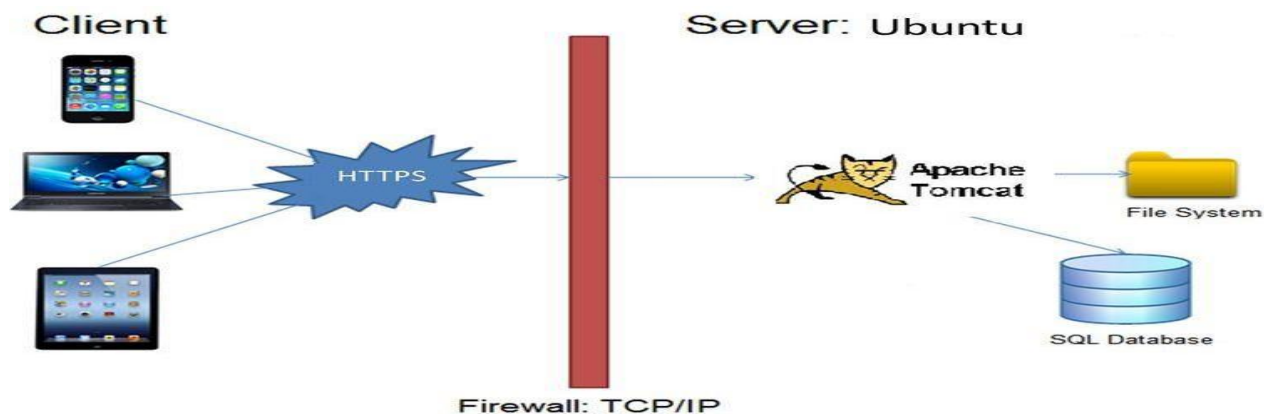


Figure 3.5-Tomcat architecture

NETBEANS IDE

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. The NetBeans

Platform is a framework for simplifying the development of Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required. Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again. NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactorings, version control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

The IDE is licensed under the Apache License 2.0. Previously, from July 2006 through 2007, NetBeans IDE was licensed under Sun's Common Development and Distribution License (CDDL), a license based on the Mozilla Public License (MPL). In October 2007, Sun announced that NetBeans would henceforth be offered under a dual license of the CDDL and the GPL version 2 licenses, with the GPL linking exception for GNU Classpath.^[16] Oracle has donated NetBeans Platform and IDE to the Apache Foundation where it underwent incubation and graduated as a top level project in April 2019.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (carries out efficient storage)
- Window management

- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library
- Integrated development tools

3.3.2 JAVA FOR ANDROID APPLICATION DEVELOPMENT

J2ME (Java 2 Platform, Micro Edition) is a technology that allows programmers to use the Java programming language and related tools to develop programs for mobile wireless information devices such as cellular phones and personal digital assistants (PDAs). J2ME consists of programming specifications and a special virtual machine, the K Virtual Machine, that allows a J2ME-encoded program to run in the mobile device. There are two programming specifications: Connected, Limited Device Configuration (CLDC) and the Mobile Information Device Profile (MIDP). CLDC lays out the application program interface (API) and virtual machine features needed to support mobile devices. MIDP adds to the CLDC the user interface, networking, and messaging details needed to interface with mobile devices. MIDP includes the idea of a midlet, a small Java application similar to an applet but one that conforms with CLDC and MIDP and is intended for mobile devices. Devices with systems that exploit J2ME are already available and are expected to become even more available in the next few years. There are a number of ways to create apps for Android devices, but the recommended method for most developers is to write native apps using Java and the Android SDK. Java for Android apps is both similar and quite different from other types of Java applications.

TOOLS REQUIRED TO DEVELOP MOBILE APPLICATION USING JAVA

The main tool required to develop android application using java is Android studio

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse

Android Development Tools (ADT) as the primary IDE for native Android application development.

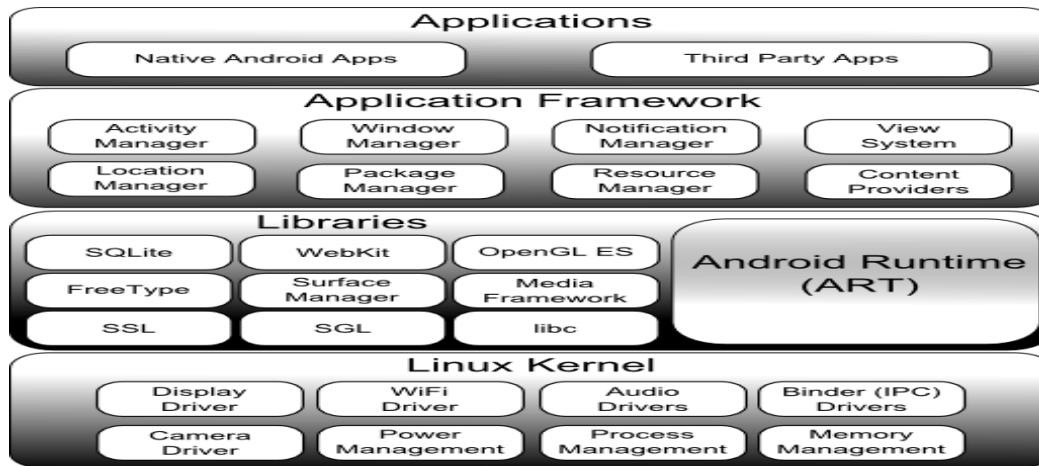


Figure 3.6-Android Architecture

CONCEPTS AND METHODOLOGIES USED

The main concepts used for building the system are

- Internet of things
- Mobile edge computing
- One-Time-Pad encryption

3.4.3 *INTERNET OF THINGS*

The Internet of Things, or IoT, refers to the billions of physical devices around the world that are now connected to the internet, all collecting and sharing data. Thanks to the arrival of super-cheap computer chips and the ubiquity of wireless networks, it's possible to turn anything, from something as small as a pill to something as big as an Airplane, into a part of the IoT. Connecting all these different objects and adding sensors to them adds a level of digital intelligence to devices that would be otherwise dumb, enabling them to communicate real-time data without involving a human being. The Internet of Things is making the fabric of the world around us smarter and more responsive, merging the digital and physical universes.

IoT Architecture

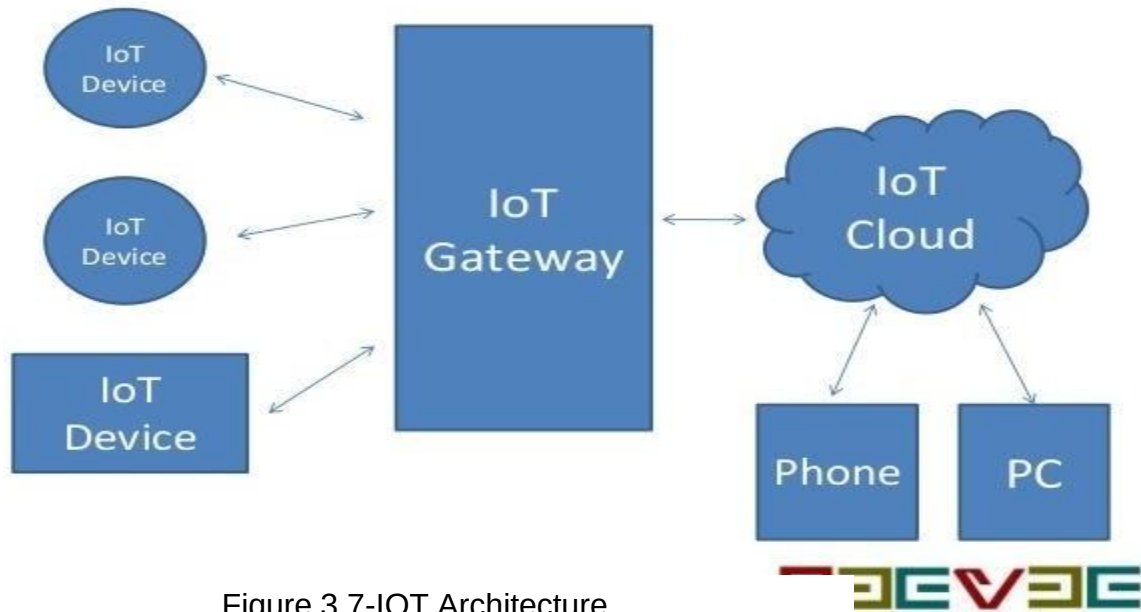


Figure 3.7-IOT Architecture

MOBILE EDGE COMPUTING

IoT Mobile Edge Computing is a combination of tiny data centers locally placed where most of the data is stored and processed locally. These network of tiny data centers are connected in the form of tree, mesh, point to point, star topologies and they push the data received to a centrally located storage repository. The range supported is within a range of 10-meter square or less. So basically in IOT Edge Computing is used for data analysis , storing and processing of data which are closer to the sensors that provide the data. Smart devices and sensors used in IoT Edge Computing are capable of processing critical data fragments and provide a quick real time response. These devices and sensors prevent the delay caused by sending the data through internet to cloud and back from cloud to local system. These devices also don't require high speed network. These devices are designed to act as small data centers that provides no latency. With this enhanced capability of Mobile edge computing, data processing is made to be implemented locally in decentralized server which effectively reduces the network traffic. This data can later be collected by the main server for further evaluation and processing. The data from the sensors or devices are sent to MEC Gateway (Mobile edge computing gateway) where the data is processed and analyzed to produce results. Mobile edge computing enhances the Cloud computing capabilities close to the Radio Access mobile Networks in 3G,

4G, 5G, LTE. CGSM, Bluetooth, ETSI is developing a system architecture and standard for a few Application program interface. Mobile edge computing manages local cache data local data breakout data analytics aggression. Virtualization in edge computing enables cloud computing providers to execute many independent application instances in a single physical server. In edge computing all the data is processed locally which is easier to manage. the user can track data and it is much secure and easy. This kind of architecture reduces complexity of implementation. The local device which process data can be made available 24/7 according to the usage of user .so we have considered to implement the system using this concept.

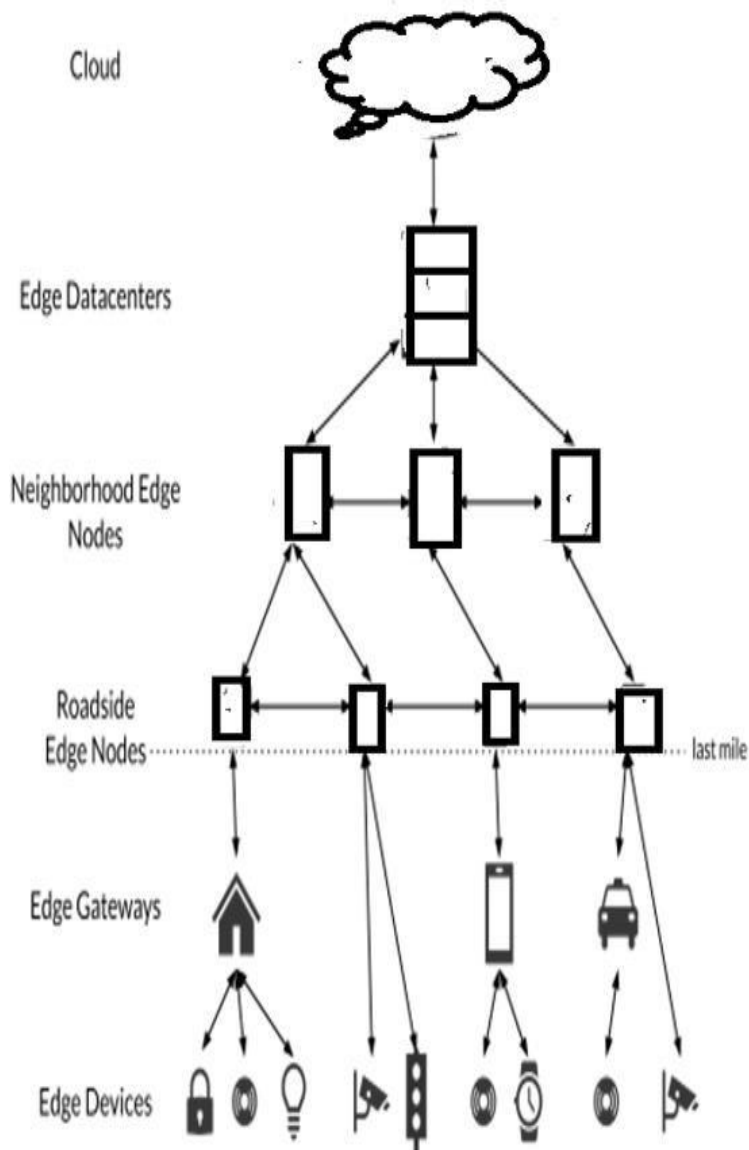


Figure 3.8-Edge computing architecture

ONE-TIME-PAD

In cryptography, the **one-time pad (OTP)** is an encryption technique that cannot be cracked, but requires the use of a one-time pre-shared key the same size as, or longer than, the message being sent. In this technique, a plaintext is paired with a random secret key (also referred to as a *one-time pad*). Then, each bit or character of the plaintext is encrypted by combining it with the corresponding bit or character from the pad using modular addition. To perform one-time pad encryption we need a key, called one-time -pad- key. The key is generated based on random technique users choose. One-time pad encryption is only possible if both sender and receiver are in possession of the same key which needs to be communicated or technique of key generation has to be known by the users. Therefore, both sender and receiver must exchange their keys beforehand and be ready. one key could be sufficient for a very long period as long as it is kept secret by users.

ENCRYPTION AND DECRYPTION

To perform encryption First covert all data to hexadecimal and store it in variable. we get the key from length of data. perform XOR operation between the data and key to get the encrypted data. Now the data is completely secure ciphertext

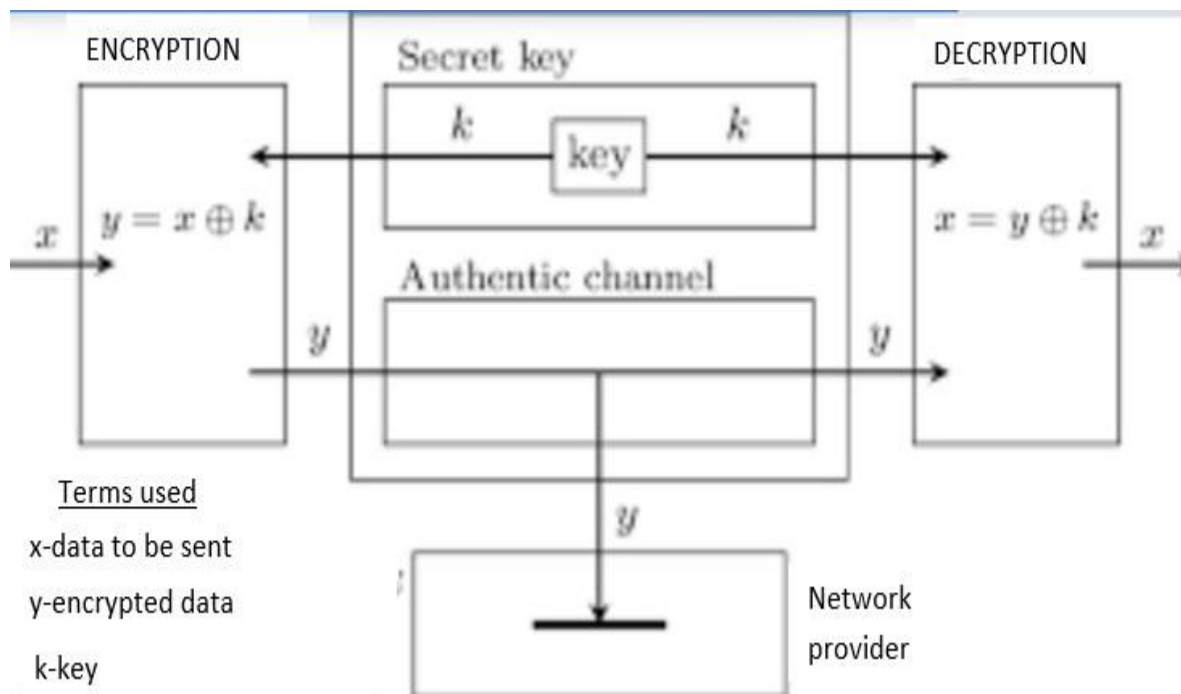


Figure 3.9-One-Time-Pad

UML DIAGRAMS

USE CASE DIAGRAM

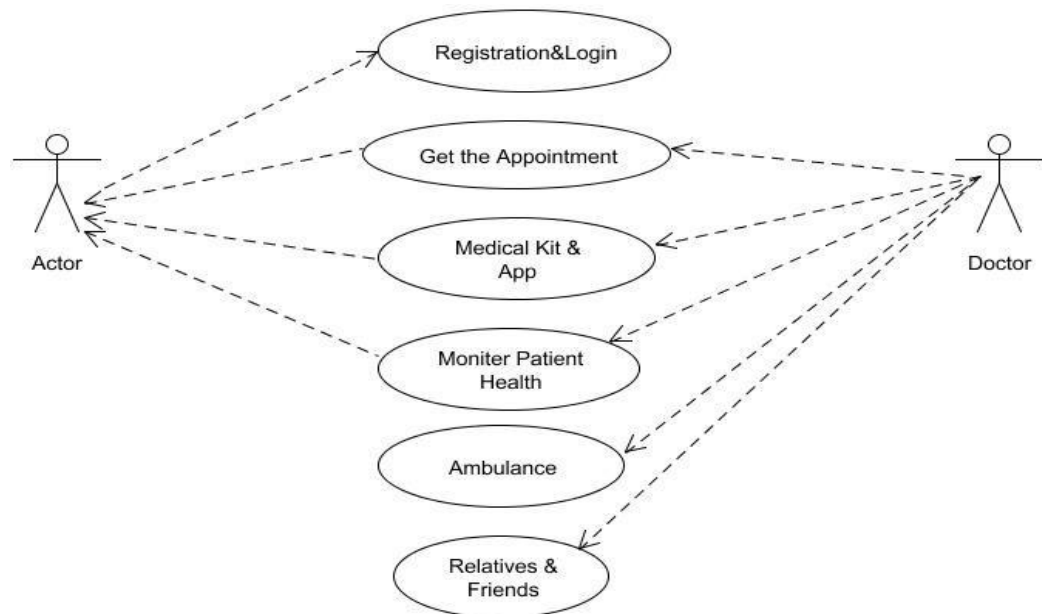


Figure-3.10-Use case diagram

DATAFLOW DIAGRAM

Level 0:

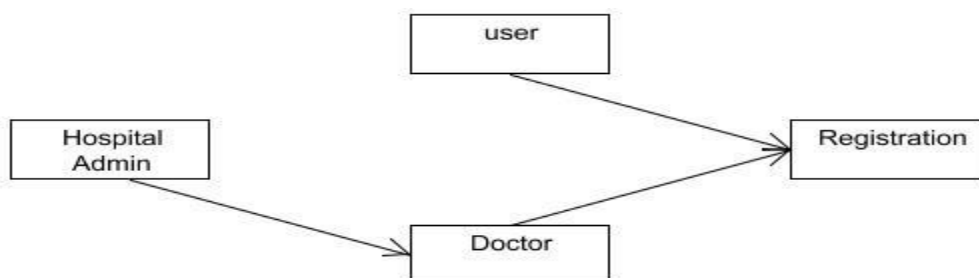


Figure-3.11 -Level 0 DFD

Level 1:

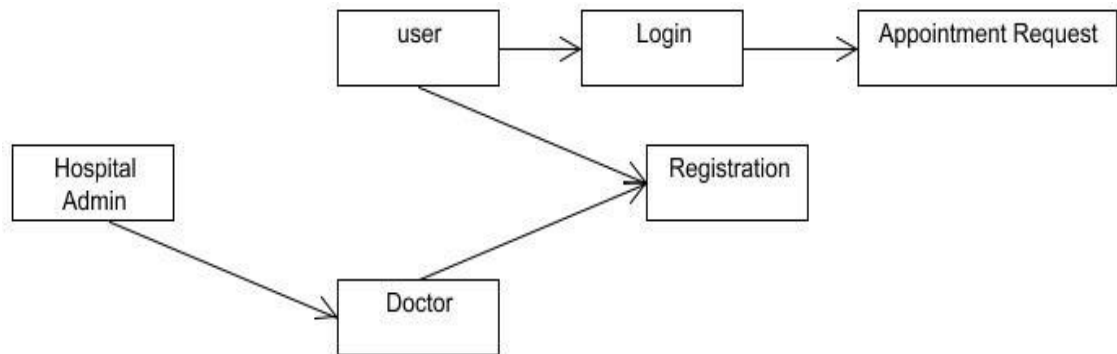


Figure-3.12-Level 1 DFD

Level 2:

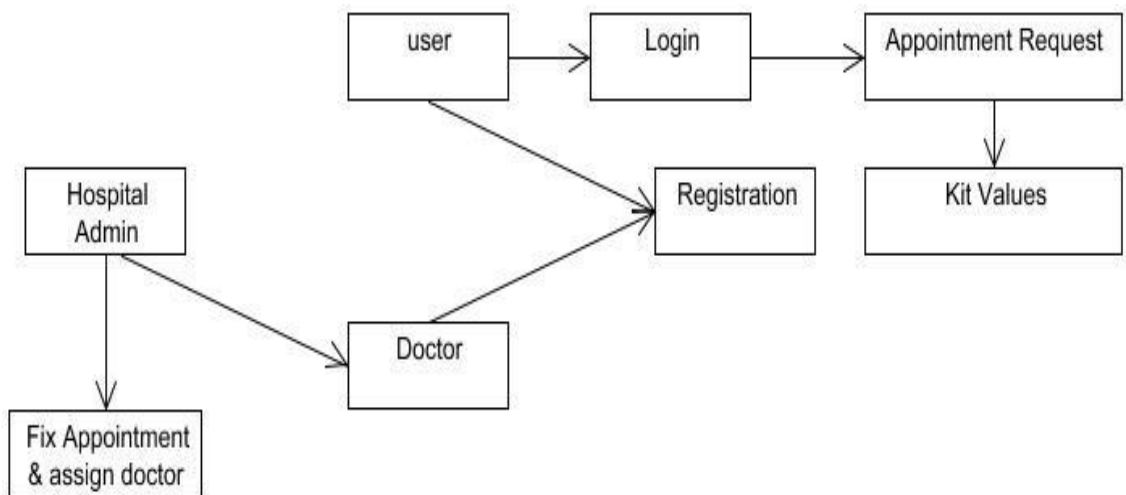


Figure-3.13-Level 2 DFD

LEVEL 3:

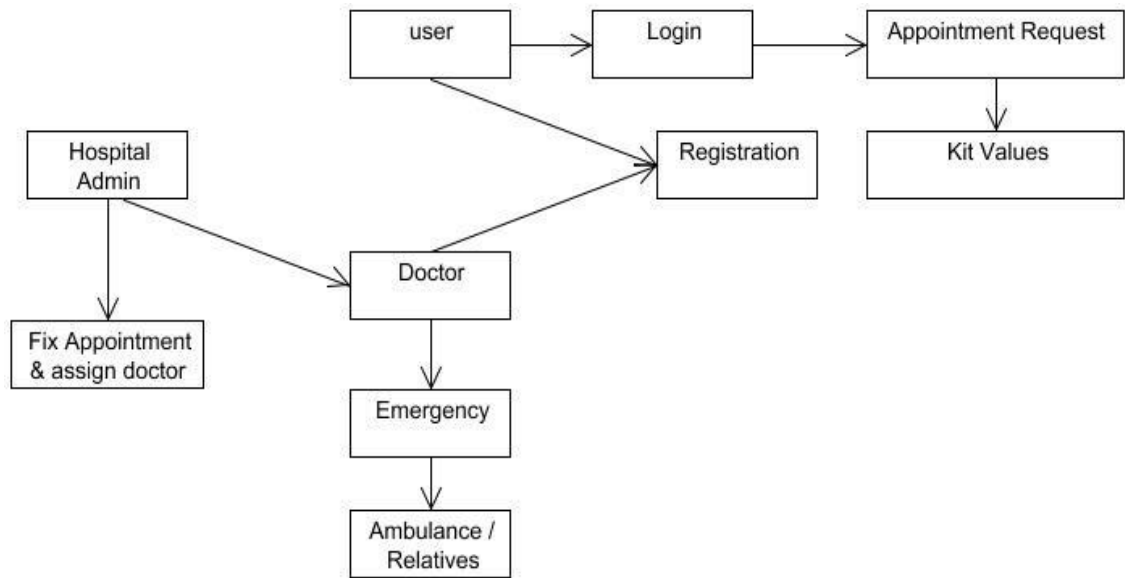


Figure-3.14 Level 3 DFD

ACTIVITY DIAGRAM

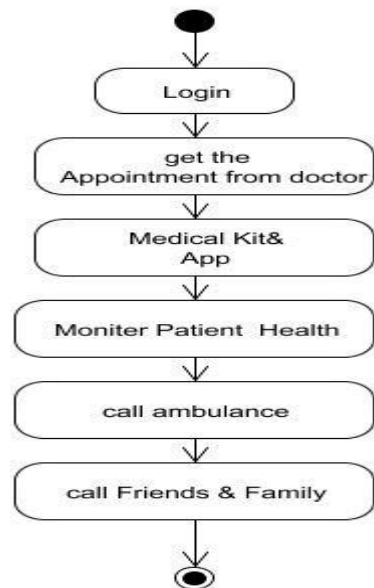


Figure-3.15-Activity diagram

3.3.3 SEQUENCE DIAGRAM

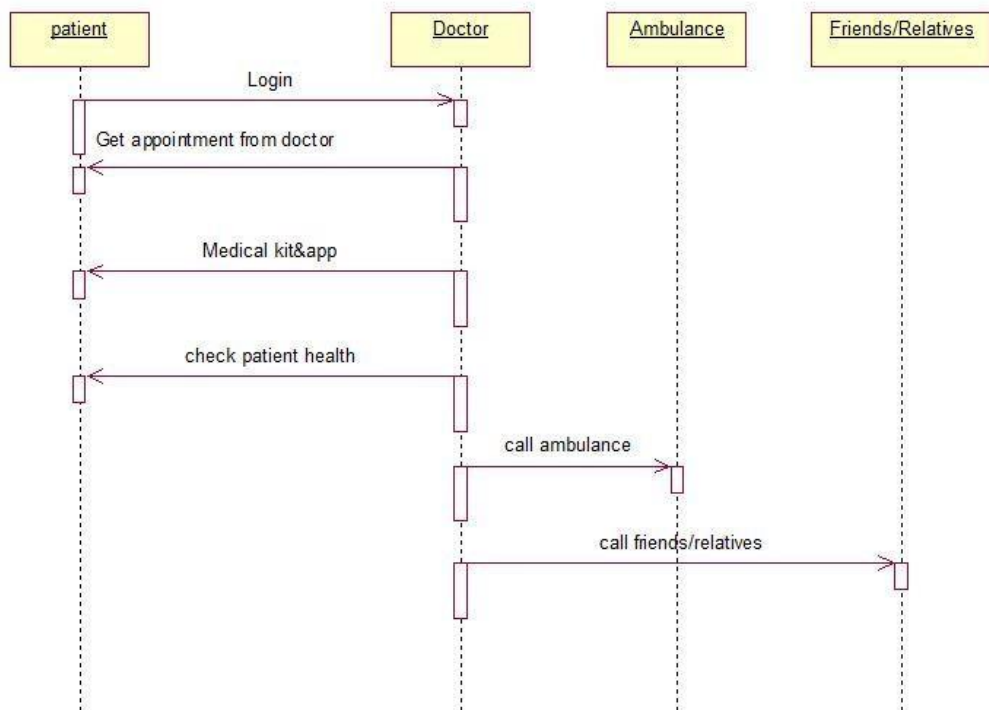


Figure-3.16-Sequence Diagram

3.5.5 COLLABORATION DIAGRAM

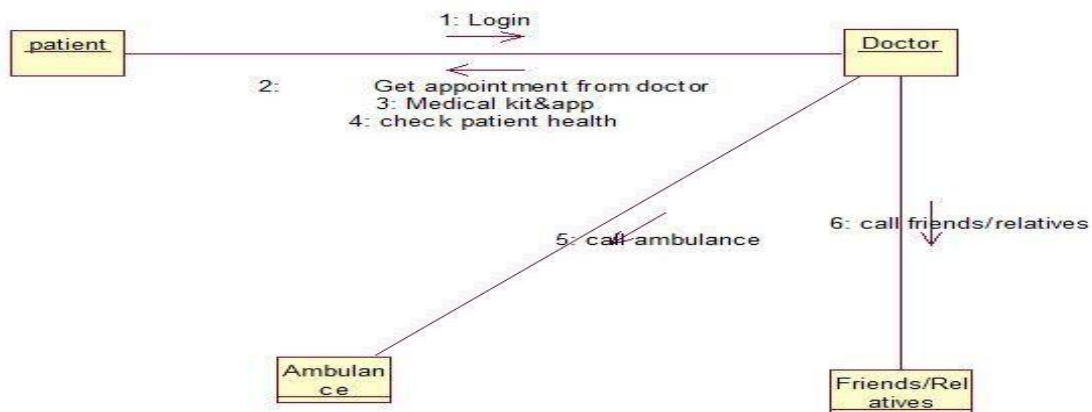


Figure-3.17-collaboration diagram

MODULES

This project is to explain the practical implementation of mobile edge computing based iot and otp encryption. We can split the project into 4 modules

1. User Authentication in Web-portal
2. Patient's Mobile Application and Appointment
3. Mobile Computation
4. Event Based Triggering

MODULE 1 USER AUTENTICATION IN WEB PORTAL:

In this module we have developed a web portal – Hospital Web application . Using this application new patients has to register their details and the data will be stored in hospital Server. Likewise a hospital Admin sign-in will be there they can add new doctors and specific specialist to which the doctors belong to . All these information will be saved into hospital server's database. We can use jsp to create web application easily.Web applications are applications that are created on request by dynamic web pages, and shall be executed on the platform provided by the Web server. Java is a kind of programming language which has one of its major characteristic as a complete independence from the physical platform on the level of translated code. Java applet and java servlet page (JSP) can be used to develop Web pages distribute them to users . The main features of java web applications are : • simple and widely accepted form of user interface (Web browser); • automatic distribution and installation of client applications; • simpler maintenance, especially in heterogeneous networks.it is recommended to code in advanced ide such as eclipse and netbeans.Because of the given features and reliability of java web development we implement the system using java.

MODULE 2 PATIENT'S MOBILE APPLICATION

In this module we create mobile application for patient this application monitors patient's health continuously and it maintains patient's health history .The patient communicates with this system by registering his/her information at first instance by answering questions related to health history and personal details. After registration, a unique identification number is provided to the patient by the cloud server. To differentiate users, cloud layer provides identification such

as patient identification (PID) and the Server also provides sets of attributes related to health history of the patients. This module works in event triggered mode. In this mode, the requisite real-time sampled data is stored at the mobile nodes. The mobile edge layer will conduct a data handling . Patients can request appointment to their respective Doctors from mobile application or web application. The mobile application can be easily developed for android using android studio using java concepts.

MODULE 3 MOBIL EDGE COMPUTATION

The health data from various sensors from respective areas will be processed in Mobile layer. We use edge-federation model of edge computing which is found to be efficient for this project implementation. Patient's own mobile device acts as the main resource for computation. Based on the previous health dataset the computation will be processed. Data from different categories will be analyzed here. Health related data from previous history got collected from the Health dataset, Environment related data like air quality, noise level around the place where patient is, Behavior related data like whether the patient is having fits, vomiting, hyper tension, fainting etc. These kinds of data get analyzed in this layer. All these data will be computed in the user's mobile phone itself. The patient will be given watch like device which will consist of set of sensors. These sensors will send the data to the patient's mobile application through mobile technology such as hotspot, bluetooth. These data are then computed in patient's phone itself. All the definitions and algorithms for the computations are defined in the patient's mobile application itself. The patient's mobile app generates a result and based on the result the application will conclude status of the patient based on critical values defined. The entire computation is controlled by the patient's mobile application. Hardware resources of the patient's phone are primarily used for computation. After computing, the end result in form of report which is in text format will be sent to the one-time-pad security mechanism.

MODULE 4 EVENT BASED TRIGGERING

This is Stage where alert SMS is sent to patient's known members and ambulance. The members to whom the SMS containing patient's health report is to be sent is decided based on the result of the patient's status. The status and health report sent will be in encrypted format. When the mobile device performs computation periodically when the values got reached above the threshold value then it will detect that patient is in emergency state and has abnormal condition. So spontaneously the mobile node will send the abnormal data to the Cloud in

encrypted format and here the Event is triggered by the cloud server. The Emergency alert to the Doctors, medical team, ambulance, relatives, etc . Thus the alert will be sent to respective mobile users. Set of actions are done according to the result from edge computation. At the receiver end the user has to click the received text and enter the key to obtain the original text. The main task of this module is to choose the members whom are supposed to receive the SMS. This module chooses the members that should receive the message based on the result .

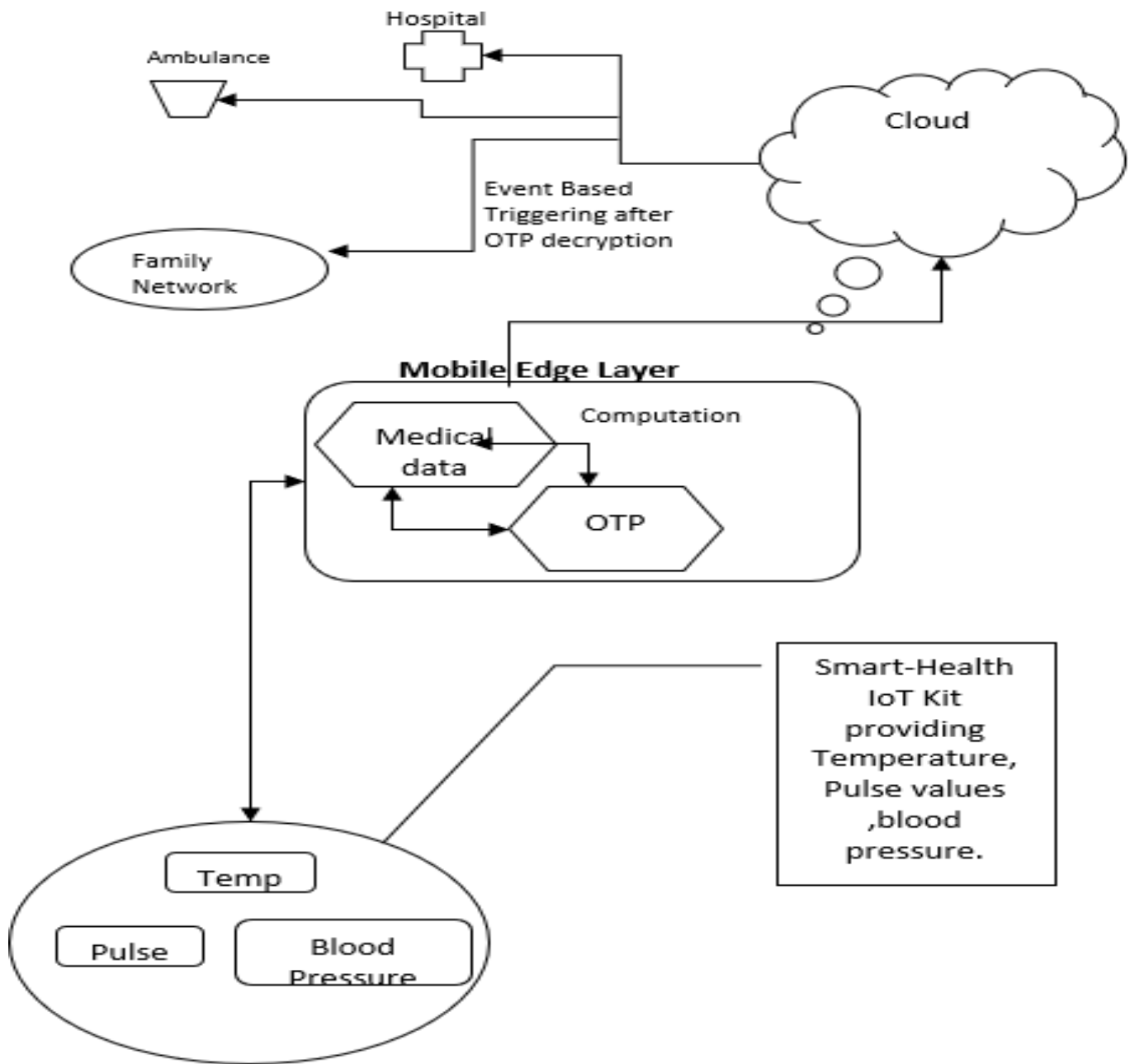


Figure-3.18-System Architecture

PROJECT HARDWARE REQUIREMENTS

COMPUTER REQUIREMENTS

- Windows 7 operating system and above
- Hard Disk: 20GB & Above
- RAM: 4GB & Above
- Processor: Pentium IV & Above

PHONE REQUIREMENTS

- Any number of phones(At least 2)
- Android: Jelly-Bean & Above
- Hard Disk : 16 Gb and above
- RAM : 2GB OR Above

HEALTH SENSING KIT REQUIREMENTS

- Temperature sensor
- Systolic and diastolic blood pressure sensor
- Pulse sensor
- Room temperature sensor
- Smoke sensor
- Arduino processor
- Bluetooth sensor

CHAPTER 4

RESULT AND DISCUSSION

WORKING

The project has four software applications to interact with

- 1.Hospital web application
- 2.Patient's mobile application
- 3.Doctor's mobile application
- 4.Ambulance management application

4.1.1HOSPITAL WEB APPLICATION

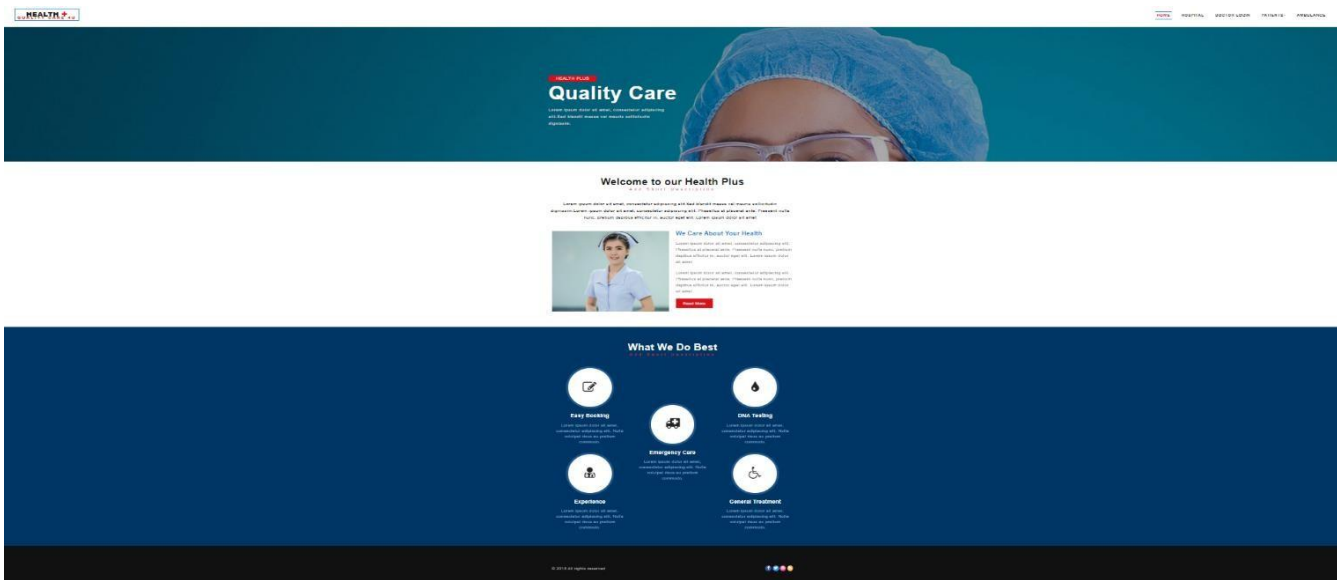


Figure 4.1 -Hospital web application

hospital web application consist of Home page, hospital admin login,Doctor Login,Patient Login and Ambulance info pages

HOSPITAL HOME PAGE

The hospital home page consist of basic details about the hospital and latest updates about the hospital.It is the page displayed when user enters hospital website.It consist of task bar with various tasks for the users to choose.

HOSPITAL ADMIN LOGIN

This is Login for hospital administrator. The hospital administrator is chosen by the hospital itself. The administrator manages every data of the website and has access to do any operation. The administrator is responsible for updating, deleting, inserting, and adding new information about the patient, doctor, ambulance. The users of the web application can also contact the admin if they face any problem by clicking contact button. When hospital task button is clicked, the following page appears for the admin to login. Admin assigns patients to doctors according to patients requirement and doctors speciality.

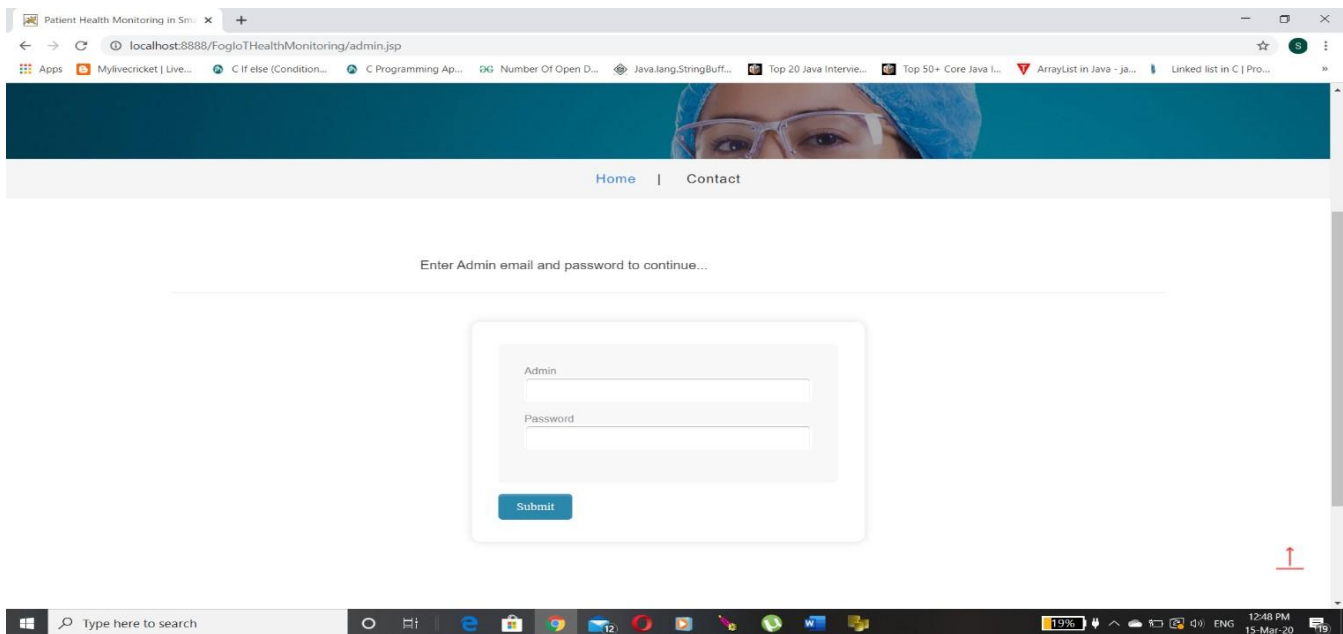


Figure 4.2-Admin signin page

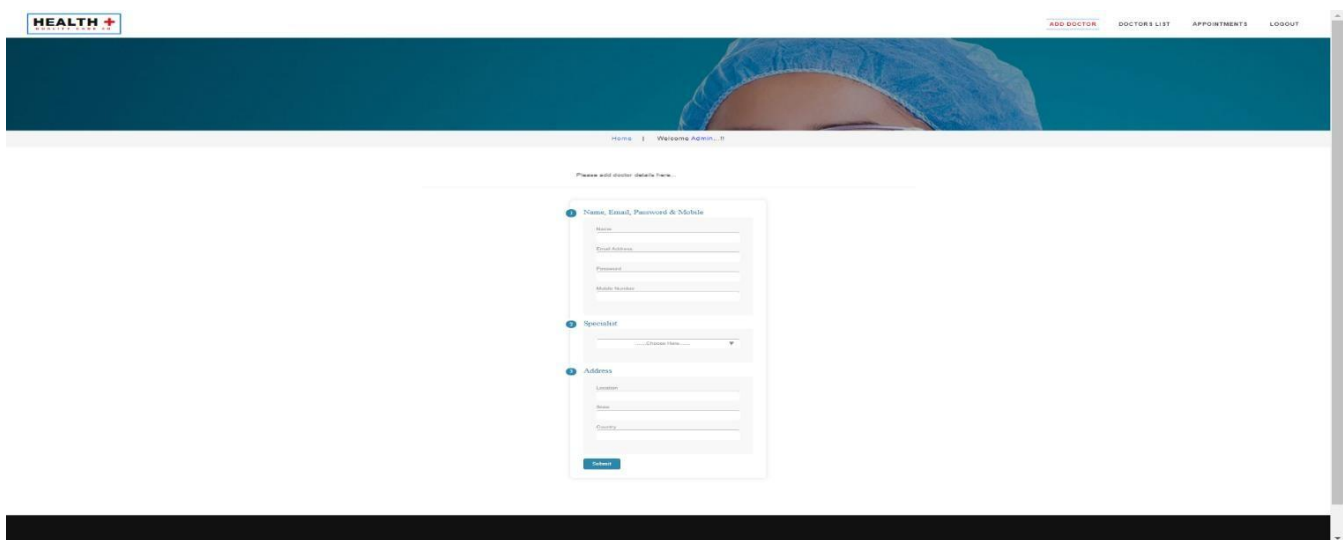
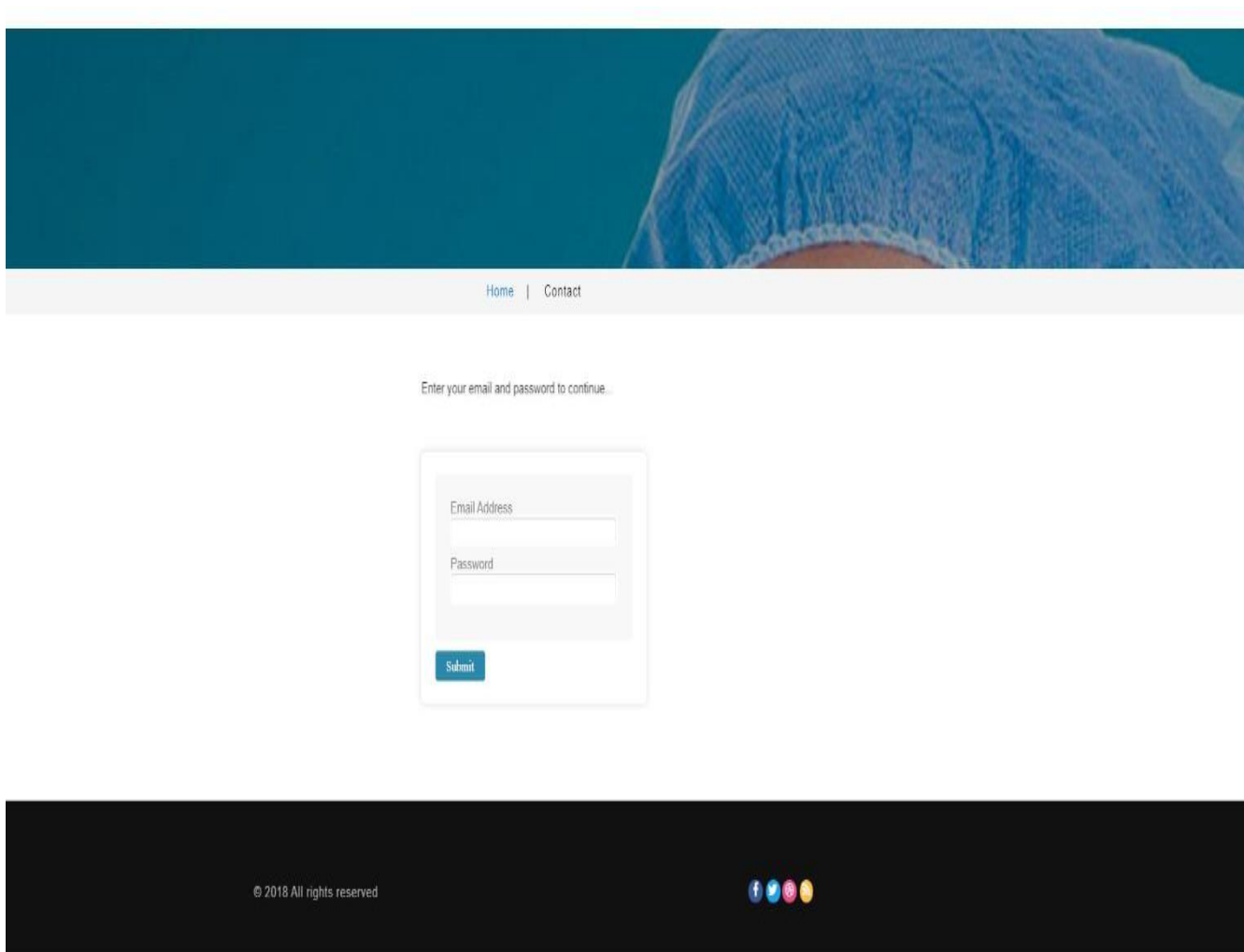


Figure 4.3-Admin control page

DOCTOR'S LOGIN

The doctors login is login for the doctors where the doctors can update his/her information. This doctors login allows the doctors to view the work assigned and various appointments fixed by the patient. This page allows the doctors to access information and medical history of patients assigned. So the doctors can know everything about the patient before meeting the patient. The doctor can contact the patient anytime using this login. The doctor will get notification of emergency cases. Patients will be categorized into abnormal critical emergency.



Home | Contact

Enter your email and password to continue.

Email Address

Password

Submit

© 2018 All rights reserved

Facebook Twitter YouTube

Figure 4.4-Doctor login page

As the doctor logs in the following page is displayed. The doctor can view the patients assigned and status of each patient and the schedule of the doctor is displayed detaildly.



Figure 4.5-Doctor's page after login

PATIENT'S TAB



Figure 4.6-Patient tab

The patients tab has two options one tab is for patients signup and another one for patient's login

PATIENT'S SIGNUP

This page is for the patient to register. Any one can register with hospital for their services by subscribing to hospital. To subscribe the patient has to sign up first. Then the patient can login and use any kind of service provided by the hospitals. To register the user has to fill all the details as shown below in the diagram.

[Home](#) | [Contact](#)

Please enter your details here

1

Name, Email, Password & Mobile

2

Age, Height, Weight & Gender

3

Emergency Intimation

4

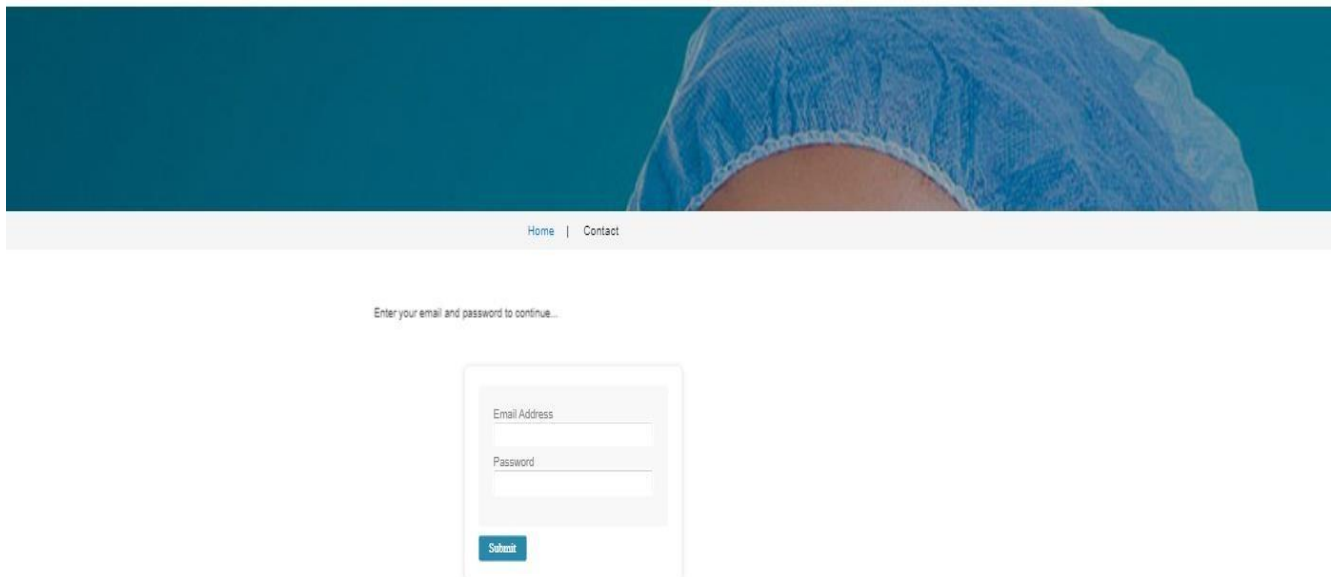
Address

Submit

Figure 4.7-Patient's Sign up

PATIENT LOGIN

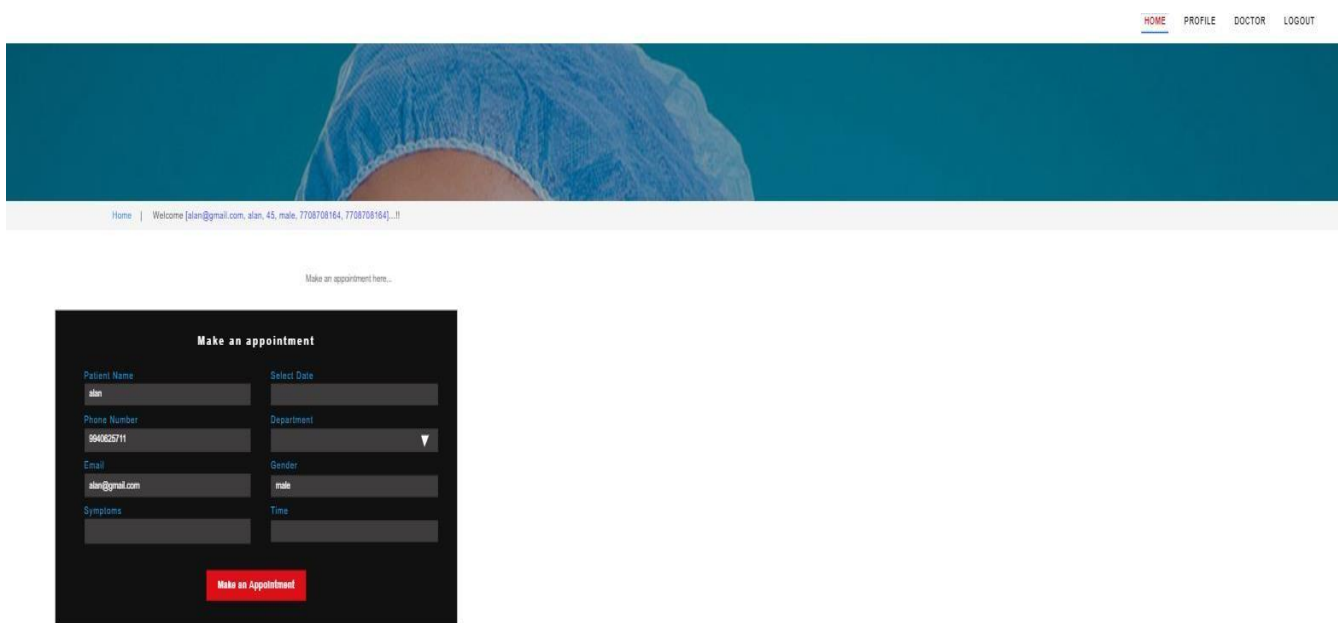
It is a page that allows patients to login. After login, a patient can access information about his/her appointments. The patient can request an appointment using this page.



The image shows a web page for patient login. At the top, there is a teal banner with a close-up of a person wearing a blue surgical cap. Below the banner is a navigation bar with links for 'Home' and 'Contact'. The main content area has a heading 'Enter your email and password to continue...' followed by a login form. The form contains two input fields: 'Email Address' and 'Password'. Below these fields is a blue 'Submit' button.

Figure 4.8-Patient's Login

Patient can enter mail and password to login .After Login the following page is displayed



The image shows a web page for making an appointment. At the top, there is a teal banner with a close-up of a person wearing a blue surgical cap. Below the banner is a navigation bar with links for 'HOME', 'PROFILE', 'DOCTOR', and 'LOGOUT'. The main content area has a heading 'Make an appointment here...' followed by a form titled 'Make an appointment'. The form contains several input fields: 'Patient Name' (with the value 'alan'), 'Select Date', 'Phone Number' (with the value '994925711'), 'Department' (a dropdown menu), 'Email' (with the value 'alan@gmail.com'), 'Gender' (with the value 'male'), 'Symptoms', and 'Time'. Below the form is a red button labeled 'Make an Appointment'.

Figure 4.9-Patient's make appointment page

After login user can view four tabs .The first tab is for requesting appointments.The second tab is profile where user can edit his or her profile .The third tab is doctor where the user can view information about the assigned doctor and the doctor's contact details.The user can logout by clicking logout button.

Please upload your image for identity purpose...

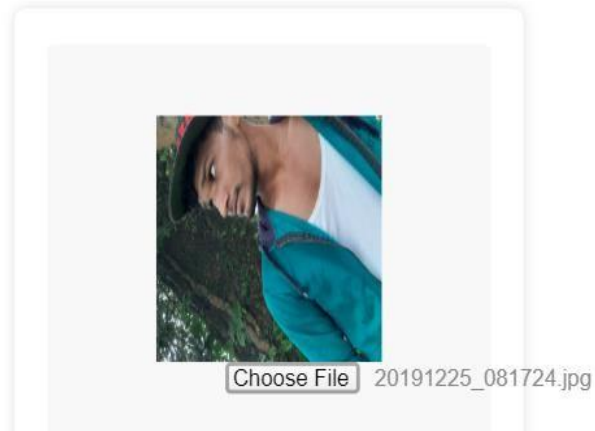


Figure 4.10-Profile page

AMBULANCE PAGE

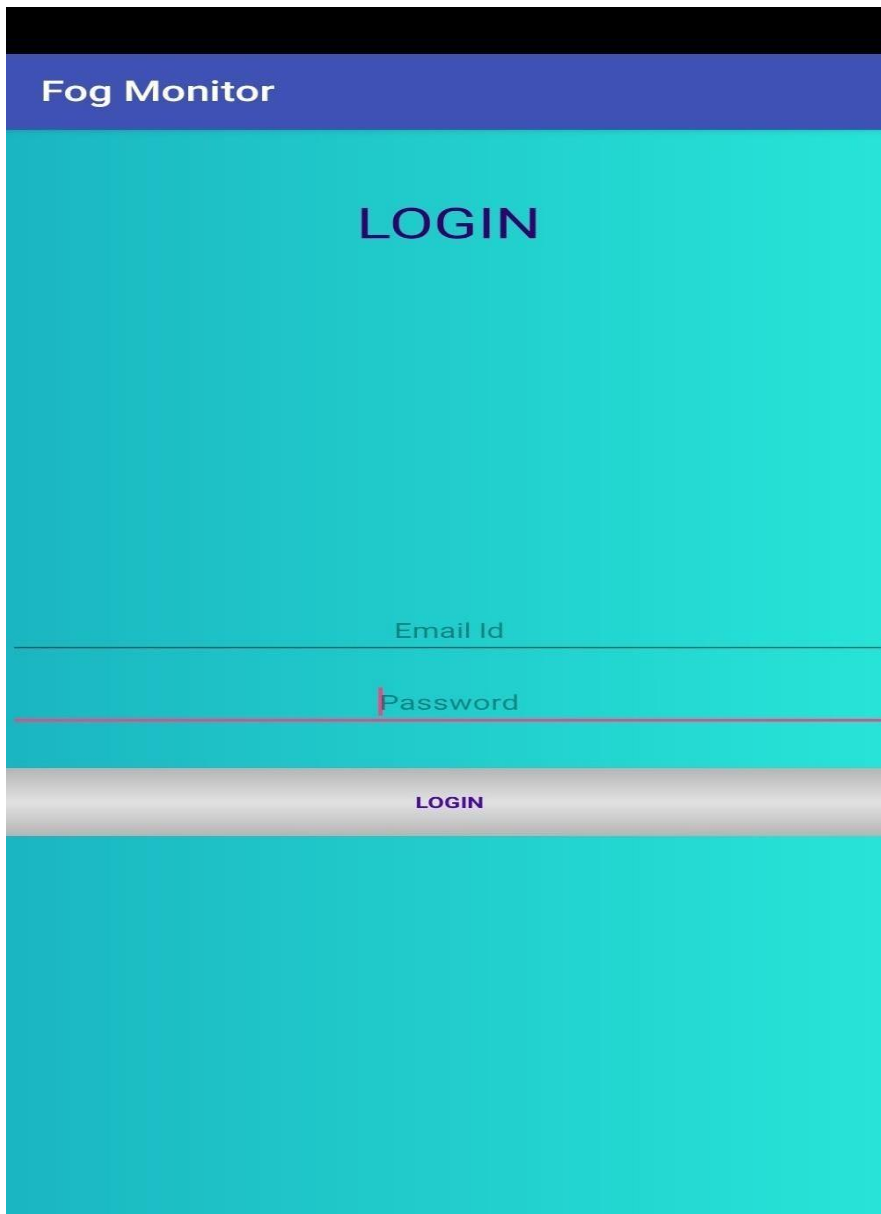
A screenshot of the 'AMBULANCE PAGE'. The page has a teal header with a close-up image of a surgical cap. Below the header is a navigation bar with links for 'Home' and 'Contact'. The main content area has the text 'Enter ambulance details here...'. Below this is a form with three input fields labeled 'Ambulance Id', 'Password', and 'Emergency Number'. At the bottom of the form is a blue 'Submit' button.

Figure 4.11-Ambulance page

In the ambulance page the hospital and user can track any ambulance by typing the ambulance id and password. The hospital admin using this page can add new ambulance and edit any existing ambulance

PATIENT'S MOBILE APPLICATION

This is mobile application version of the patient's tab in we server. Whatever patient can do in the web application patient can do in the mobile application also .The patient can fix appointment , view doctors contact details, update profile ,view health history and so on.



The image shows a mobile application interface for 'Fog Monitor'. At the top, there is a black header bar. Below it is a blue header bar with the text 'Fog Monitor' in white. The main background is a light blue gradient. In the center, the word 'LOGIN' is displayed in large, bold, dark blue capital letters. Below 'LOGIN', there are two input fields: 'Email Id' and 'Password'. The 'Email Id' field has a thin black underline, and the 'Password' field has a thin red underline. Below these fields is a grey button with the word 'LOGIN' in dark blue capital letters. The bottom of the screen is a solid light blue area.

Figure 4.12-Patients mobile app

The patient can also fix appointment by clicking fix appointment button .To fix appointment first user have to login the user have to click appointment button the fill in the details and click fix appointment button to fix appointment .

Fog Monitor

Make an Appointment

Patient Name
CNU

Gender
male

Phone Number
9940625711

Email
cnustr@gmail.com

Symptoms
heart attack

Select Date
31/01/20

Department
select

Select Time
05:00 AM

FIX APPOINTMENT

Figure 4.1-Make appointment through app

The patient's mobile application continuously collects data from sensors for a time period chosen by the patient and it continuously processes the data to produce the result. The mobile application performs the mobile edge computations. Data from the sensors is fed to the application through Bluetooth. This application also encrypts the result and sends SMS.

Fog Monitor

Kit Values

Systolic	100
Diastolic	80
Pulse	65
Body Temperature	98
Room Temperature	41

SEND **APPOINTMENT**

Figure 4.14-Kit values processing

The values from the sensors are fed into the appropriate columns as shown in the above figure Then the status of the patient is computed.

DOCTOR'S MOBILE APPLICATION

This is mobile application version of the Doctor's tab in we server.Whatever Doctor can do in the web application patient can do in the mobile application also.Using this application the doctor can monitor all the patients digitally.In this project The patient's data is collected through sensors and sent to main server .The main server allows access of patients data to the doctor so the doctor can monitor the patient online at any time.

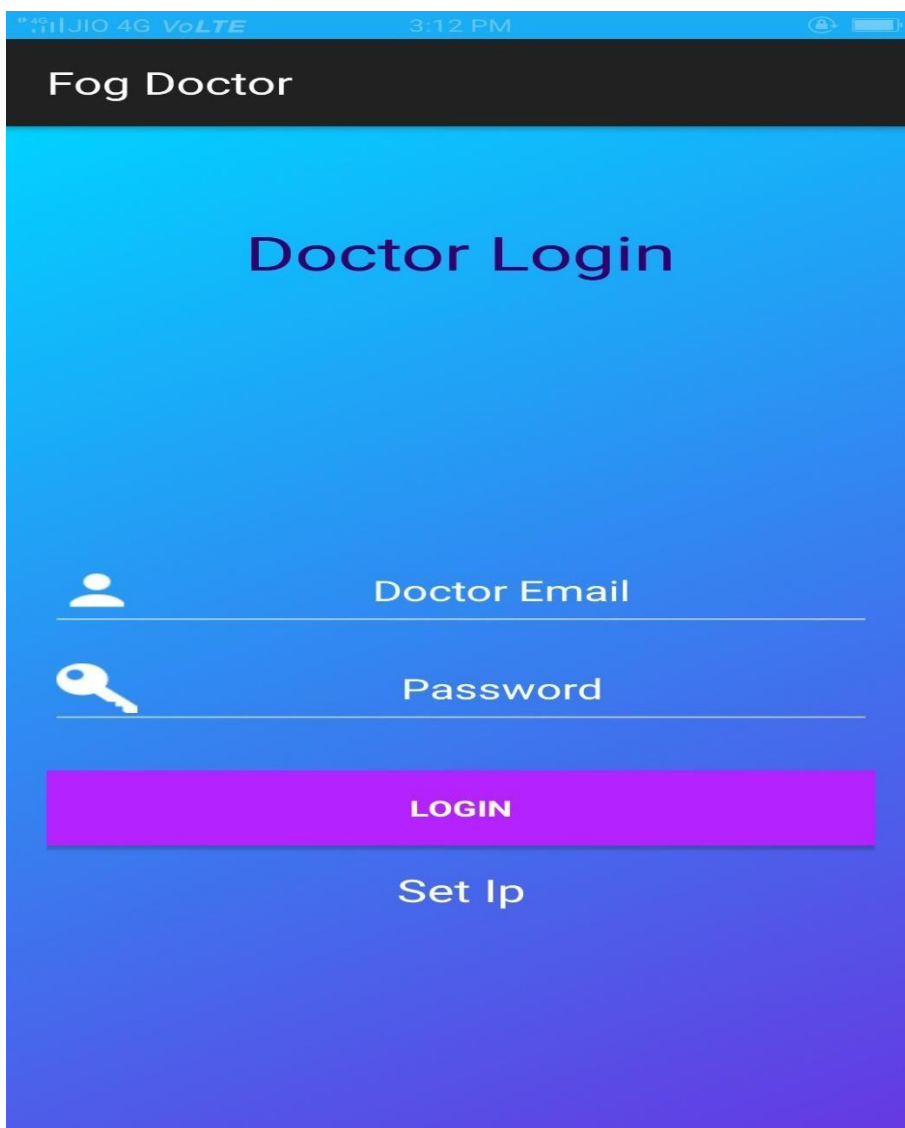


Figure 4.15-Doctor login

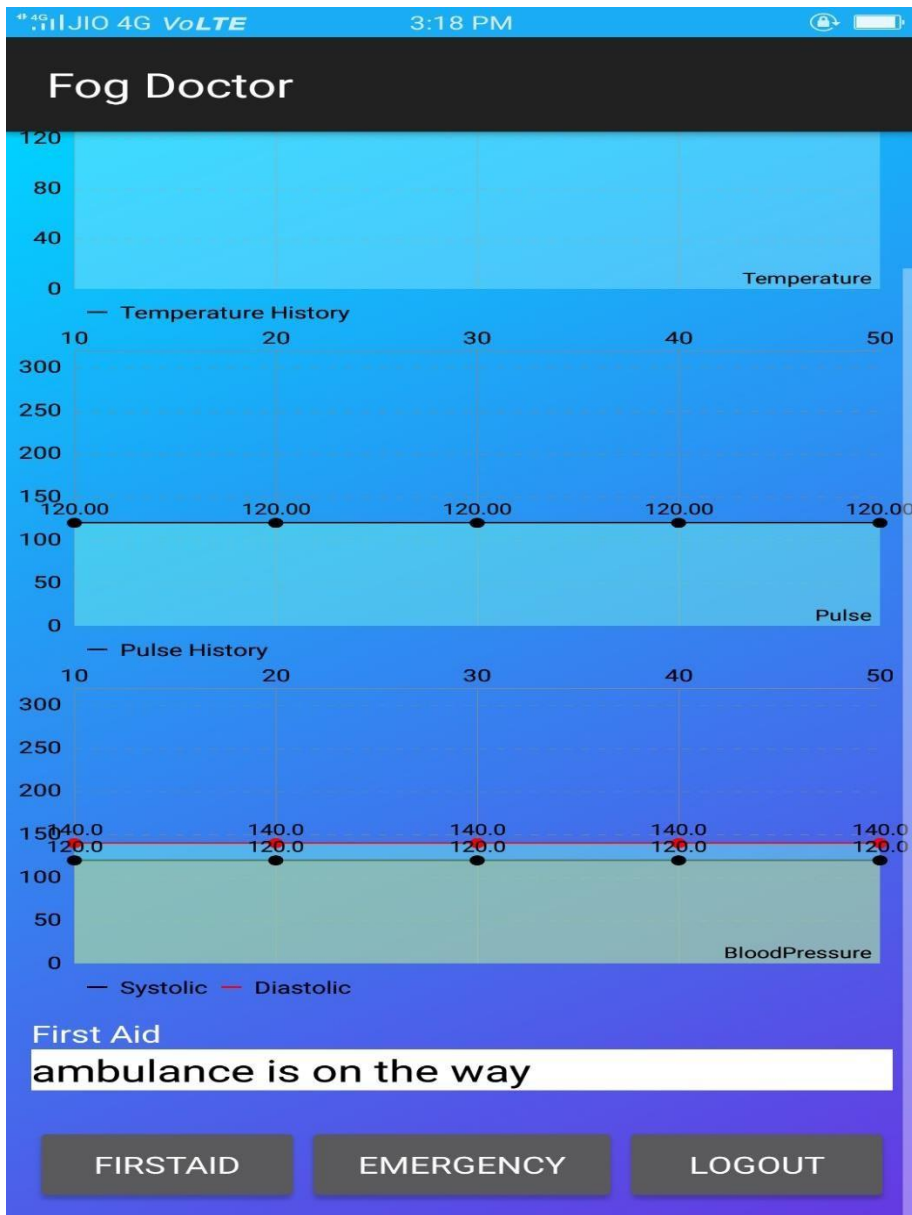


Figure 4.16- patient report

Doctor can login and view any patient's data and can send any message to the patient. If any emergency or critical situation The doctor and hospital immediately gets notification and the doctor can view the critical report of patient.

AMBULANCE APPLICATION

The ambulance application is for ambulance drivers which gives notification and location about the patients that are in emergency. The hospital admin also can track ambulance using this application.

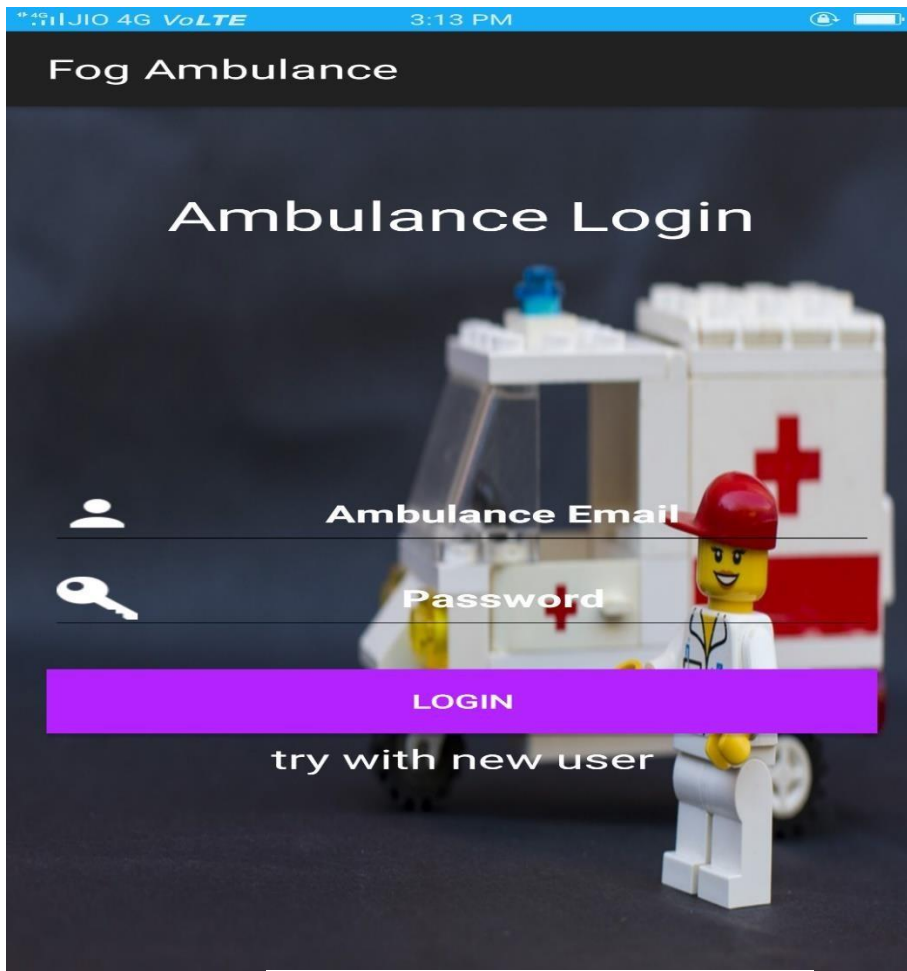


Figure 4.17-Ambulance Login

RESULTS

The result of the project is divided into 4 categories as mentioned below

1. Normal condition
2. abnormal condition
3. critical condition
4. emergency condition

NORMAL CONDITION

The application computes the status of patient based on sensor values. When the values are normal the application displays result as normal. In this state no SMS is sent to any known members of the patient. The result is displayed only to the patient.

ABNORMAL CONDITION

This is when the health values are little higher or lower than normal values. This means the patient needs to visit doctor within 24 hours and needs to take extra care and the medications properly.

CRITICAL CONDITION

This is when the values are too higher or too lower than normal values. An ambulance is arranged for the patient to visit the hospital immediately. The nearest ambulance is arranged. And sms is sent to all the members including neighbor, close relatives.

EMERGENCY CONDITION

In this ambulance along with experienced doctor along with all the required medical equipment and medications are immediately sent to patient's location And SMS is sent to all known members.

CHAPTER 5

SUMMARY AND CONCLUSION

CONCLUSION

Hence in this project we have evaluated the sustainability and efficiency of edge computing using mobile networks. Specifically, we evaluated the performance of edge computing in health care application as an example system to understand the mobile edge concepts in depth. The end results have shown that edge computing is necessary for efficient less resource oriented and fail free system in health application and other cases. Although regional data centers allow to significantly reduce network latency, only hosting resources at the edge enables a satisfactory quality of experience for proposed system. Furthermore, increasing the computational capabilities of the servers in the cloud does not compensate for the increase in network latency. Therefore, deploying even limited computing resources at the edge helps improve the quality of experience in the considered system. We hope that this article will encourage further research in this direction and evolve the edge computing technology further.

REFERENCES

BOOKS

1. Algorithm Design ,Book by Jon Kleinberg and Éva Tardos
2. An introduction to object-oriented programming, Book by Timothy Budd
3. Core Java Volume I – Fundamentals BY Cay S. Horstmann
4. Core Java: An Integrated Approach , Book by R. Nageswara Rao
5. Core Java: Fundamentals, Book by cay s horstmann
6. Data Structures & Algorithms in Java , Book by Robert Lafore
7. Effective Java 2nd Edition, Book by Joshua-Bloch
8. Effective java by Joshua Bloch
9. Effective Project Management in easy steps: 2nd edition, Book by John M.Carroll
10. Java How to program: Sixth Edition, Book by H.M.Deitel and P.J.Deita
11. Murach's Java Servlets and JSP, 3rd Edition by kevin jhones

LINKS

1. <https://www.javatpoint.com › jsp-tutorial>

APPENDIX

A. SAMPLE CODE:

CODE FOR WEB APPLICATION ADMIN PAGE

```
package controller;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.http.servlet.HttpServlet;

public class Admin extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        HttpSession session1=request.getSession(true);

        String email=request.getParameter("email");

        String password=request.getParameter("password");

        if(email.equals("admin@gmail.com") && password.equals("admin"))

        {session1.setAttribute("namesession", "Admin");

            RequestDispatcher rd=request.getRequestDispatcher("dregister.jsp");

            rd.forward(request, response);}

        else

            request.setAttribute("msg", "Invalid Email or Password!");

            RequestDispatcher rd=request.getRequestDispatcher("admin.jsp");

            rd.forward(request, response);
```

```

    }

    out.close();

}

```

DOCTOR LOGIN PAGE

```

package controller;import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.commons.fileupload.FileItem;

import org.apache.commons.fileupload.FileItemFactory;

import org.apache.commons.fileupload.FileUploadException;

import org.apache.commons.fileupload.disk.DiskFileItemFactory;

import org.apache.commons.fileupload.servlet.ServletFileUpload;

import service.DoctorDao;

import com.http.servlet.HttpServlet;

public class DoctorProfile extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        HttpSession session1=request.getSession(true);

        String demail=(String)session1.getAttribute("demail");

        String image="";

        System.out.println("Doctor email is----->" +demail);

```

```

boolean isMultipart = ServletFileUpload.isMultipartContent(request);

FileItemFactory factory = new DiskFileItemFactory();

ServletFileUpload upload = new ServletFileUpload(factory);

try {String filepath = "";List<FileItem> items = upload.parseRequest(request);

    for (FileItem locFileItem : items)

        {if (locFileItem.isFormField())

            {System.out.println("---"+locFileItem.getFieldName());

                }else

            {filepath = "webapps/FogIoTHealthMonitoring/Identity/"+demail;

                String path=filepath

                File f = new File(path);

                if (!f.exists()) {

                    f.mkdirs();}

                image = locFileItem.getName();

                System.out.println("Filename is " + image);

                File tosave = new File(path, image);

                try {locFileItem.write(tosave);

                    } catch (Exception e) {

                        e.printStackTrace();

                    }

            }

        boolean status=DoctorDao.setDoctorProfile(demail, image);

        if(status==true)

        {

            RequestDispatcher rd=request.getRequestDispatcher("adminmenu.jsp");

            rd.forward(request, response);

            System.out.println("image uploaded");

        }

        else

        {RequestDispatchererrd=request.getRequestDispatcher("dprofile.jsp");

```

```

        rd.forward(request, response);

        System.out.println("image not uploade}

    }catch (FileUploadException e) {

        e.printStackTrace();

    }

    out.close();}}

```

PATIENT LOGIN PAGE

```

package controller;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import service.PatientLogService;

import com.http.servlet.HttpServlet;

public class Patientog extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        HttpSession session1=request.getSession(true);

        String email=request.getParameter("email");

        String password=request.getParameter("password");

        System.out.println("email is ---->" +email);

        boolean echeck=PatientLogService.patientEmailCheck(email);

        if(echeck==true)

            {String status=PatientLogService.loginPatient(email, password);

```

```

        if(status!="no")

        {System.out.println("Login successful ----->" +status);

                session1.setAttribute("namesession", status);

                session1.setAttribute("emailsession", email);

//                request.setAttribute("msg", "");

                RequestDispatcher rd=request.getRequestDispatcher("appointment.jsp");

                rd.forward(request, response);}else

        System.out.println("Login failed");

                request.setAttribute("msg", "Invalid Patient Password!");

                RequestDispatcher rd=request.getRequestDispatcher("plogin.jsp");        rd.forward(request, response)}}

        else

        {System.out.println("Email not exists in a database");

                request.setAttribute("msg", "Invalid Patient Email");

                RequestDispatcher rd=request.getRequestDispatcher("plogin.jsp");

                rd.forward(request, response);

        }

        out.close();

    }}

```

PATIENT FIX APPOINTMENT PAGE CODE

```

package controller;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.List;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

```

```

import model.AppointmentPojo;

import service.PatientDao;import com.http.servlet.HttpServlet;

public class FixAppointment extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

response.setContentType("text/html");                PrintWriter out = response.getWriter();

        String pemail=request.getParameter("pemail");

        String pdate=request.getParameter("pdate");

        String ptime=request.getParameter("ptime");

        System.out.println("Patient email ----->" +pemail);

        List<AppointmentPojo> list=PatientDao.getPatientAppointmentDetails(pemail, pdate, ptime);

        ArrayList<AppointmentPojo> al=new ArrayList<AppointmentPojo>();

        al.addAll(list);

        request.setAttribute("al", al);

        RequestDispatcher rd=request.getRequestDispatcher("fixappointment.jsp");

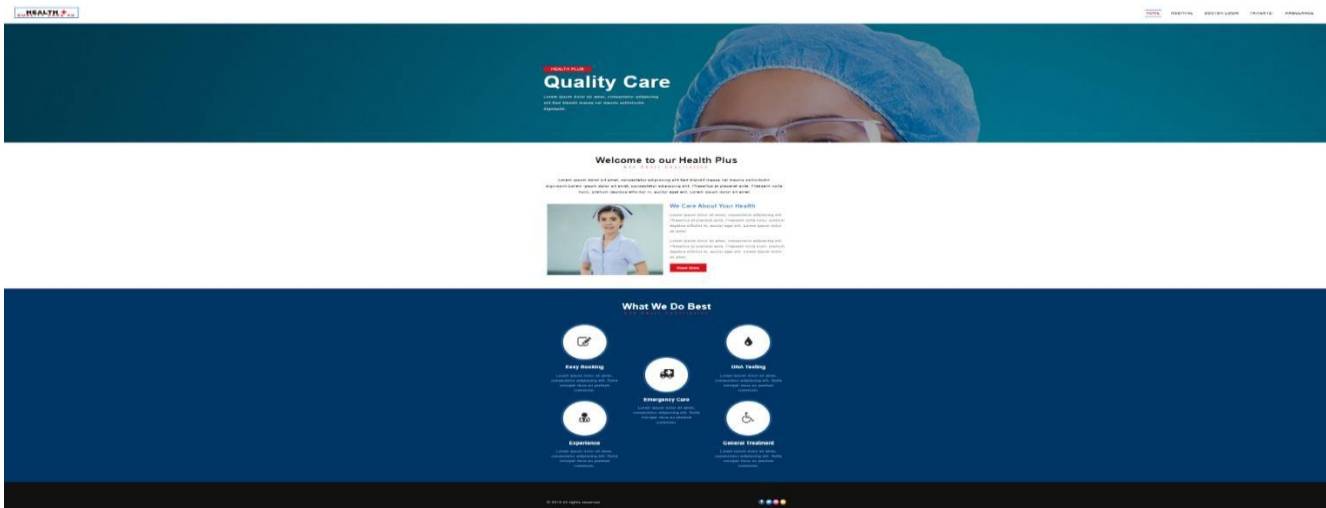
        rd.forward(request, response);

        out.close();}}

```

B. SCREENSHOTS

WEBPAGE



ANDROID APP FOR PATIENT

The screenshot shows the 'Fog Monitor' app interface for patient login. It features a dark blue header with the app name. The main area has a light blue background with the word 'LOGIN' in large, bold, black letters. Below this, there are two input fields: 'Email Id' and 'Password'. The 'Password' field has a red vertical bar on the left side. At the bottom, there is a dark blue button labeled 'LOGIN'.

Fog Monitor

LOGIN

Email Id

Password

LOGIN

ANDROID APP FOR DOCTOR

The screenshot shows the 'Fog Doctor' app interface for doctor login. It features a dark blue header with the app name. The main area has a light blue background with the text 'Doctor Login' in large, bold, black letters. Below this, there are two input fields: 'Doctor Email' and 'Password'. The 'Doctor Email' field has a person icon on the left, and the 'Password' field has a key icon on the left. At the bottom, there is a dark blue button labeled 'LOGIN' and a link labeled 'Set Ip'.

Fog Doctor

Doctor Login

Doctor Email

Password

LOGIN

Set Ip

ANDROID APP FOR AMBULANCE TRACKING



REAL TIME WORKING



C. PUBLICATION PAPER

MOBILE EDGE COMPUTING BASED IOT WITH ONE TIME PAD ENCRYPTION

V.Shivadhar

V.Srinivasan

Prathima Devadas

ABSTRACT

We are in a period where Cloud platform and edge computing are becoming a major platform for the computation, processing of various data and communication between machines, sensors and smart objects, in real time applications and for many crucial tasks. With IOT expanding in various real time applications various techniques of implementing IOT are Emerging. It is important to understand the use of Edge computing which will provide methodology for information sharing, data processing, data storing and data analyzing. The applications of Edge computing in IoT is keep on expanding with improving technology in day to day life. This Paper is to assist and help researchers working in the area of the Edge computing platform to understand concepts involved in edge computing easily. This paper also explains standard instructions to be followed on how to protect messages with one-time pad encryption easily in simplest way and how to implement mobile edge computing based IOT with least use of resources.

KEY WORDS: MOBILE EDGE COMPUTING, ONE TIME PAD ENCRYPTION, SECURITY

INTRODUCTION

Computers and electronic gadgets are required whenever rate of correctness, automation, speed and accurate computations are needed. In the conventional computing system, the input is given by human beings and the software applications running at the systems processes the inputted data and produces an accurate output as possible. generally, the outputs are displayed on a output device such as a monitor or printer. As humans are lazy and there is possibility of inaccuracy in humans IOT systems are used. IOT Systems are 100 percent accurate and it eliminates possibility of human errors. The processing application is located at a central server in the case of an IoT system. The sensors in the IOT systems forward their data to a computing system via variety of networks such as mobile networks, Bluetooth, hotspot, WIFI, transmitter and receiver. The local computing system preprocesses all the data it obtains and sends it to a central server continuously using networks. In IOT an actuator or event-based notification system is used. Thus, an IoT system uses sensors, actuators and processors together to generate output. So IOT helps in improving automation in industries and cooperate companies. so we can summarize an IoT-enabled system is set of sensors, embedded system, electronics and communication devices which implements data storage via Internet into a cloud server and uses an analytics software to make decisions at the cloud platform uses help of analytics system to make certain decisions which is sent back to local device or end user. At periodic intervals, the collected data are analyzed using analytics software running in the server (cloud). The decisions and results from server are then informed accordingly to the end user. In most cases, it turns out to be the notification in the mobile device or actuation of some subsystems of the end device. We can use various technologies and concepts along

with IOT. IOT can be implemented together with edge computing, fog computing, mobile computing and cloud

computing or combination of these concepts. IN this paper we are going to see Different ways of implementing IOT with example system.

RELATED WORK

Wireless sensor networks consist of resource constrained devices that sense, process and transmit data from their surroundings to main server . They are launched in places that can be reached by humans. Due to their nature, many applications are available for their use such as health, military, monitoring and detecting abnormality in various cases[12]. Due to their computational speed, computational power and transmission constraints, the efficient use of sensors and resource constrained devices are essentially important. The mobile based communication, for example, places the highest strain on the energy conservation of the nodes [13]. Many related work has been done using edge computing concept in field of virtual reality and games[5] ,communication and networks[10] ,image processing[23].

EXISTING SYSTEM

IoT based remote monitoring systems are suggested by many professionals because of their reliability, time saving properties, accuracy and speed. In existing system, the IoT devices will keep observe the data from various resources and send data to the respective fog nodes for computation of health data. Then the computed values will be transferred to the cloud Server. Based on the abnormal condition the Cloud will send intimation to the clients. Use of fog nodes is not that efficient

as it makes the network complex and inefficient. This result in massive storage accumulation in cloud storage, processing the collected data is obviously a hard task to perform which results in computational overhead and delay in sending emergency alert.

PROPOSED SYSTEM

We propose the Secured OTP IOT Environment in Medical Field. Here the patient's android mobile device will be acting as the edge server for doing computing with the receiving data from health monitoring devices. The health-related data get uploaded into Mobile node. In Mobile Edge Layer, edge level computing is performed based on the threshold values the health status of respective patient is analyzed. If the status is getting abnormal then the Mobile Edge layer sends the health data with One-time pad security to the Cloud. Then the emergency SOS message will be sent immediately to ambulance, friends, close relatives, neighbors based on the type of event triggered. We propose in this system that the SMS is sent even without internet connectivity as it is for lifesaving and emergency purpose. Mobile computing promotes highly efficient output because maximum computation is done edge level which makes the cloud process as light weight. This system comes over the disadvantages of existing system and increase efficiency and decrease network and software requirements.

COMPARISON BETWEEN FOG COMPUTING, CLOUD COMPUTING AND MOBILE EDGE COMPUTING IN IOT.WHICH IS MORE EFFICIENT FOR PROPOSED SYSTEM?

IMPLEMENTATION USING CLOUD

Cloud provides data storage and hardware services virtually where the processing hardware is in random location. In cloud computing the data is sent to some data processor located in virtual location Where the data is processed and the result is sent back to the main server or the user application. This requires high speed internet and cloud service provider for cloud server which is time consuming and another problem is data maybe lost or easily hacked during tranctaction.For the proposed system high speed internet may not be available and it is required that in less time data is processed so implimenting using cloud is not much efficient .Availability is another problem with cloud. For the proposed system we need server available for 24/7 .In cloud we may have to send the data to randomly located long distance server in which there may be data loss during transfer and because of long distance data tranfer there are security issues .Data may be tampered with during tranfer .So cloud model is not considered for the proposed system.Cloud have security issues also.

IMPLEMENTATION USING FOG COMPUTING

Fog is a kind of distributed networking model that is closely associated with cloud computing and the internet of things. In fog computing multiple fog nodes are created each nodes are given individual data and each node process data individually and the data are combined and result is generated . Use of fog nodes is not that efficient as it makes the network complex and inefficient. This result in massive storage accumulation in cloud storage, processing the collected data is obviously a hard task to perform which results in computational overhead and delay in sending emergency alert. The huge disadvantage with fog computing is it cannot handle huge quantity of data continuously .Implimentation and maintance of fog is complex.Comparing with requirements of our proposed system implementing using fog will not be much efficient .

IMPLEMENTATION USING MOBILE EDGE COMPUTING

IoT Mobile Edge Computing is a combination of tiny data centers locally placed where most of the data is stored and processed locally. These network of tiny data centers are connected in the form of tree, mesh, point to point, star topologies and they push the data received to a centrally located storage repository. The range supported is within a range of 10-meter square or less. So basically, Edge Computing is used for data analysis, storing and processing of data which are closer to the sensors that provide the data. Smart devices and sensors used in IoT Edge Computing are capable of processing critical data fragments and provide a quick real time response. These devices and sensors prevent the delay caused by sending the data through internet to cloud and back from cloud to local system. These devices also don't require high speed network. These devices are designed to act as small data centers that provides no latency. With this enhanced capability of Mobile edge computing, data processing is made to be implemented locally in decentralized server which effectively reduces the network traffic. This data can later be collected by the main server for further evaluation and processing. The data from the sensors or devices are sent to MEC Gateway (Mobile edge computing gateway) where the data is processed and analyzed to produce results. Mobile edge computing enhances the Cloud computing capabilities close to the Radio Access mobile Networks in 3G, 4G, 5G, LTE. CGSM, Bluetooth, ETSI is developing a system architecture and standard for a few Application program interface. Mobile edge computing manages local cache data local data breakout data analytics aggression. Virtualization in edge computing enables cloud computing providers to execute many independent application instances in a single physical server. In edge computing all the data is processed locally which is easier to manage. the user can track data and it is much secure and easy. This kind of architecture reduces complexity of implementation. The local

device which process data can be made available 24/7 according to the usage of user. The most important benefit of mobile edge computing is its ability to increase network performance by reducing latency. Since IoT edge computing devices process data locally or in nearby edge data centers, the information they collect doesn't have to travel nearly as far as it would under a traditional cloud

Approach	Edge nodes	Edge network
cloudlets	Compact size datacenters deployed on WIFI access	Wifi,3G,LTE
Mobile cloudlets	Compact size datacenters on cars	3G LTE
Multi access edge computing	Servers deployed in radio servers	3G,LTE,Other access technologies
Fog computing	Heterogeneous nodes including high end routers, access point setup box	Multiple wireless access technologies including WIFI,3G,LTE
Mobile cloud	Neighboring mobile nodes form a cloud with one device chosen as resource coordinator	Local networking through WIFI Bluetooth internet connectivity with wifi,3G ,LTE
Edge cloud	Compute or storage nodes deployed in edge network and federated to cloud data centers	Home/enterprise networks and WIFI hotspots
fusion	service nodes deployed on access points, local and centralized data centers	Not defined

Table 1- Different approaches in Edge

CLASSIFICATION OF EDGE COMPUTING PLATFORMS

There are three classifications (A)resource rich servers that are deployed at the edge level , (b) heterogeneous edge based computing nodes and (c) edge to cloud federation. The classification is based on how edge is deployed. However, in practice, one category can be used by combining one category with others categories. One category relies on servers with rich set of resources that are deployed very close(below 100 feet) to the end-devices. Another group uses the resources from various nodes present at the edge, including the end devices themselves. A third category is based on the federation of resources at the edge nodes and centralized data centers. each category is explained below in detail

A)RESOURCE RICH SERVERS DEPLOYED AT EDGE:

One option to realize an edge computing platform is to deploy the resource-rich servers in the network to which end-users are connected. A cloudlet is defined as a set of data centers that

architecture. Mobile edge computing could manage heavy network traffic so it is suitable. So we have considered to implement the system using this concept.

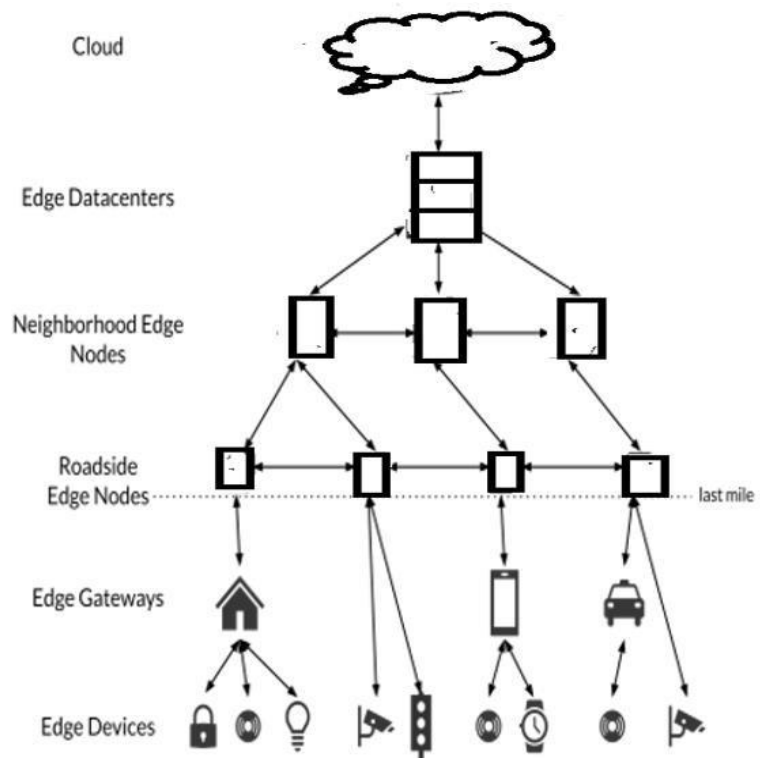


Figure 1-Edge computing architecture

offers sets of clusters of multicore computing units, data storage and wireless LAN connectivity towards the edge. All the sensor data collected from sensors are processed on the cloudlet to provide real-time assistance and analysis assistance. Since the cloudlets are launched and introduced, further research professionals have proposed integration of cloudlets with femtocells, the LTE base stations define the architecture of micro data centers consisting of thousands of servers which are capable of hosting interactive applications for end-users. While these data centers have been used for deploying Content Delivery networks(CDNs) and email applications. they can also be reengineered to host cloudlets. Similar to cloudlets which will be deploying a small set of servers on WiFi access points or a base station in the radio access network is called as a micro cloud.

B)HETEROGENEOUS EDGE NODES:

In contrast to the set of techniques and solutions that are described above, edge computing platforms can produce a

diverse set of computing resources. It consists of a fog platform devices (including smartphones and connected vehicles). Heterogeneity of wireless connectivity is a key aspect of the end-devices. Thus, many kinds of wireless access network(WAN) technologies are supported by their solution.fog computing is a cooperation based mobile cloud, wherein heterogeneous mobile devices opportunistically share their resources to deliver particular and appropriate services and applications which are characterized by a highly virtualized system of heterogeneous nodes, ranging from resource-rich servers to more constrained edge routers, access points, set-top boxes and even end .

C) EDGE CLOUD FEDERATION MODEL :

Another option to understand edge platforms is based on the combination of computing resources at the edge and centralized data centers describe this concept as an edge cloud. In this system, edge applications are used to deliver services and resources at the edge as well as in distant cloud centers. Many professionals advice the use of edge apps to launch indoor 3D localization and video monitoring applications. Similarly, the correct combination of private and public clouds enable integrated IoT applications. In this Type of edge computing system, the edge node continuously orchestrates the federation of nodes to maximize the number of tasks that are executed. For the fusion architecture in this model federation of cloud is important factor . In this edge model,the services are deployed on a cloud infrastructure which are then distributed throughout the Internet.

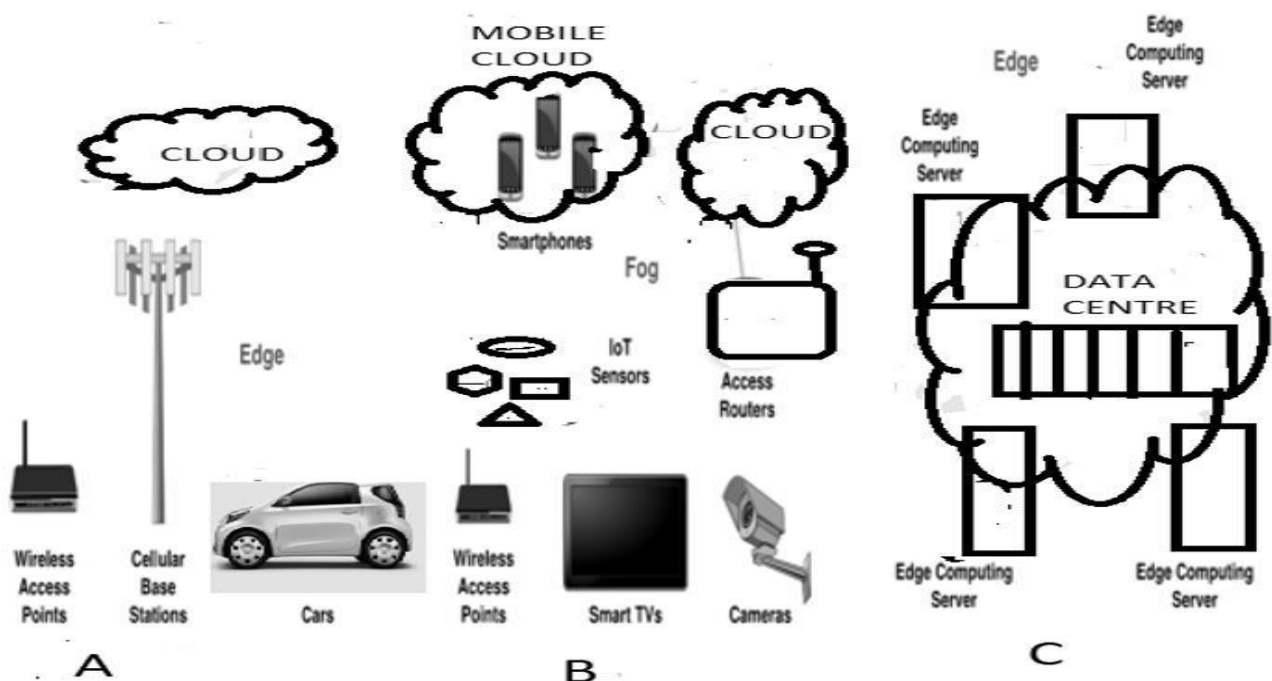


Figure 2 classification of edge models A)Resource rich edge server B)Heterogeneous edge nodes C)Edge-cloud federation

MONITORING HEALTH DATA USING EDGE BASED IOT WITH OTP

This project is to explain the practical implimentation of mobile edge computing based iot and otp encryption. We can split the project into 4 modules

1. User Authentication in Web-portal
2. Patient's Mobile Application and Appointment
3. Mobile Computation
4. Event Based Triggering

MODULE 1 USER AUTENTICATION IN WEB PORTAL:

In this module we have developed a web portal – Hospital Web application . Using this application new patients has to register their details and the data will be stored in hospital Server. Likewise a hospital Admin sign-in will be there they can add new doctors and specific specialist to which the doctors belong to . All these informations will be saved into hospital server's database. We can use jsp to create web application easily. Java is a kind of programming language which has one of its major characteristic as a complete independence from the physical

platform on the level of translated code. Java applet and java servlet page (JSP) can be used to develop Web pages distribute them to users . Using jsp we combine major java concepts such as jdbc, spring, hibernate to create web server. The main features of java web applications are : it is recommended to code in advanced ide such as eclipse and netbeans. Because of the given features and reliability of java web development we implement the system using java.

MODULE 2 PATIENT'S MOBILE APPLICATION

In this module we create mobile application for patient this application monitors patient's health continuously and it maintains patient's health history .The patient communicates with this system by first registering his/her information at first instance by answering few questions related to health history and personal details. After registration is successfully done, a unique number for identification purpose is provided to the patient by the cloud server. To differentiate users, cloud layer provides identification such as patient identification (PID) and the Server also provides sets of attributes related to health history of the patients. This module works in event triggered mode. In this mode, the requisite real-time sampled data is stored at the mobile nodes. The mobile edge layer will conduct a data handling . Patients can request appointment to their respective Doctors from mobile application or web application. The mobile application can be easily developed for android using android studio using java concepts.

MODULE 3 MOBILE EDGE COMPUTATION

The health data from various sensors from respective areas will be processed in Mobile layer. We use edge-federation model of edge computing which is found to be efficient for this project implementation. Patient's own mobile device acts as the main resource for computation. Based on the previous health dataset the computation will be processed. Data from different categories will be analyzed here. Health related data from previous history got collected from the Health dataset, Environment related data like air quality, noise level around the place where patient is, Behavior related data like whether the patient is having fits, vomiting, hyper tension, fainting etc. These kinds of data get analyzed in this layer. All these data will be computed in the user's mobile phone itself. The patient will be given watch like device which will consist of set of sensors these sensors will send the data to the patient's mobile

application through mobile technology such as hotspot, bluetooth. These data are then computed in patient's phone itself. All the definitions and algorithms for the computations are defined in The patient's mobile application itself .The patient's mobile app generates a result and based on the result the application will conclude status of the patient based on critical values defined .The entire computation is controlled by the patient's mobile application .Hardware resources of the patient's phone are primarily used for computation . After computing , the end result in form of report which is in text format will be sent to the one-time-pad security mechanism. The one time pad encryption and decryption will be implemented as given below.

ONE TIME PAD ENCRYPTION

One-time pad encryption known as most secure and stable method to protect short text messages. This paper explains how to implement one-time pads, and how to deal with its various security issues. Working with one-time pads is easy to learn. The one-time -pad system is transparent and we don't need any computer, special equipment or any knowledge about cryptography techniques or mathematics .to implement one time pad encryption . The system will surely provide unbreakable encryption when properly used and implemented. It won't be possible to crack one-time pad encrypted data without having the proper key, even as future attack technology may evolve .The one-time pad (OTP), also called as Vernam cipher or the perfect cipher, is a cryptographic algorithm where plain text is combined with a random key to produce an encrypted data. The OTP is uncrackable only if four important rules are followed. When rules are applied accurately, the one-time pad can be proven unbreakable. However, if any one of the given rules is not satisfied, the cipher is no longer unbreakable. The following

1. The minimum length of key is as long as the plaintext.
2. The technique of key generation is kept between the users.
3. There must only be 2 copies of the key: one key for the sending user and another for the receiving user (some exceptions exist for multiple receivers).
4. The keys are to be used only once, and both sending user and receiving must destroy each of their keys after use

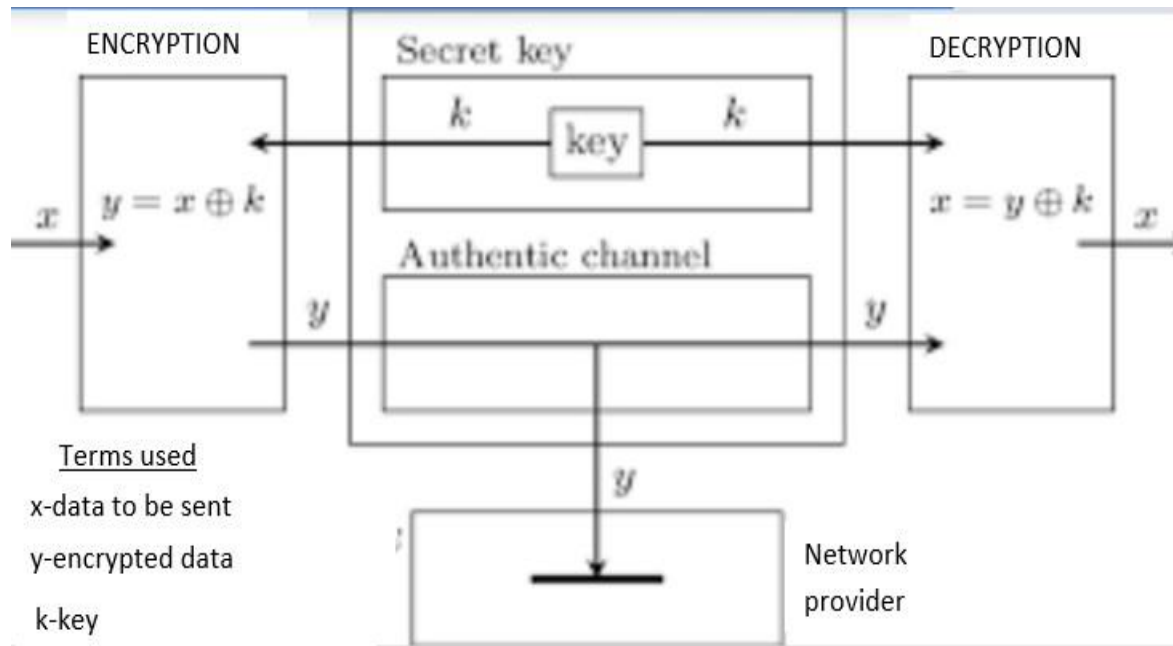


Figure 3-One time pad diagrammatic representation

IMPLEMENTING ONE TIME PAD ENCRYPTION

To perform one-time pad encryption we need a key, called one-time -pad- key. The key is generated based on random technique users choose. One-time pad encryption is only possible if both sender and receiver are in possession of the same key which needs to be communicated or technique of key generation has to be known by the users. Therefore, both sender and receiver must exchange their keys beforehand and be ready. one key could be sufficient for a very long period as long as it is kept secret by users.

ENCRYPTION

To perform encryption First covert all data to hexadecimal and store it in variable. we get the key from length of data. perform XOR operation between the data and key to get the encrypted data. Now the data is completely secure ciphertext! Let us see an example with medical data

Bp 110

Pulse 65

1) hexadecimal form of above data

4270313130

50756c7365203635

2)key(length=26)

3236

3)encrypted data

4270310306

50756c7365200403

the message is sent by radio, in voice or Morse, or by telephone, or SMS .it is recommended to relay all groups twice to avoid problems.

DECRYPTION

Write the one-time pad digits underneath the ciphertext and add ciphertext .Perform operations from encryption in reverse order . Calculate key from encrypted data. Here the technique used to generate key is nothing but finding the length of data therefore the key is equal to length of data perform reverse xor operation of encrypted data with key to get final data. Let's see an example.

1)encrypted data

4270310306

50756c7365200403

2)key calculation

Length=26=3236

3)decryption(reverse xor)

4270313130

50756c7365203635

4)hexadecimal to text

Bp 100

Pulse 65

After decryption the receiving user can successfully view the message .

We create the encryption in the application itself . The application automatically encrypts the data .We are create the encryption algorithm using arithmetic and bitwise operators in java concepts.The

MODULE 4 EVENT BASED TRIGGERING

This is Stage where alert SMS is sent to patient's known members and ambulance.The members to whom the SMS containing patients health report is to be sent is decided based on the result of the patient's status.The status and health report sent will be in encrypted format. When the mobile device performs computation periodically when the values got reached above the threshold value then it will detect that patient is in emergency state and has abnormal condition. So spontaneously the mobile node will send the abnormal data to the Cloud in encrypted format and here the Event is triggered by the cloud server. The Emergency alert to the Doctors, medical team, ambulance, relatives, etc . Thus the alert will be sent to respective mobile users. Set of actions are done according to the result from edge computation.At the receiver end the user has to click the received text and enter the key to obtain the original text. The main task of this module is to choose the members whom are supposed to receive the SMS.This module chooses the members that should receive the message based on the result .

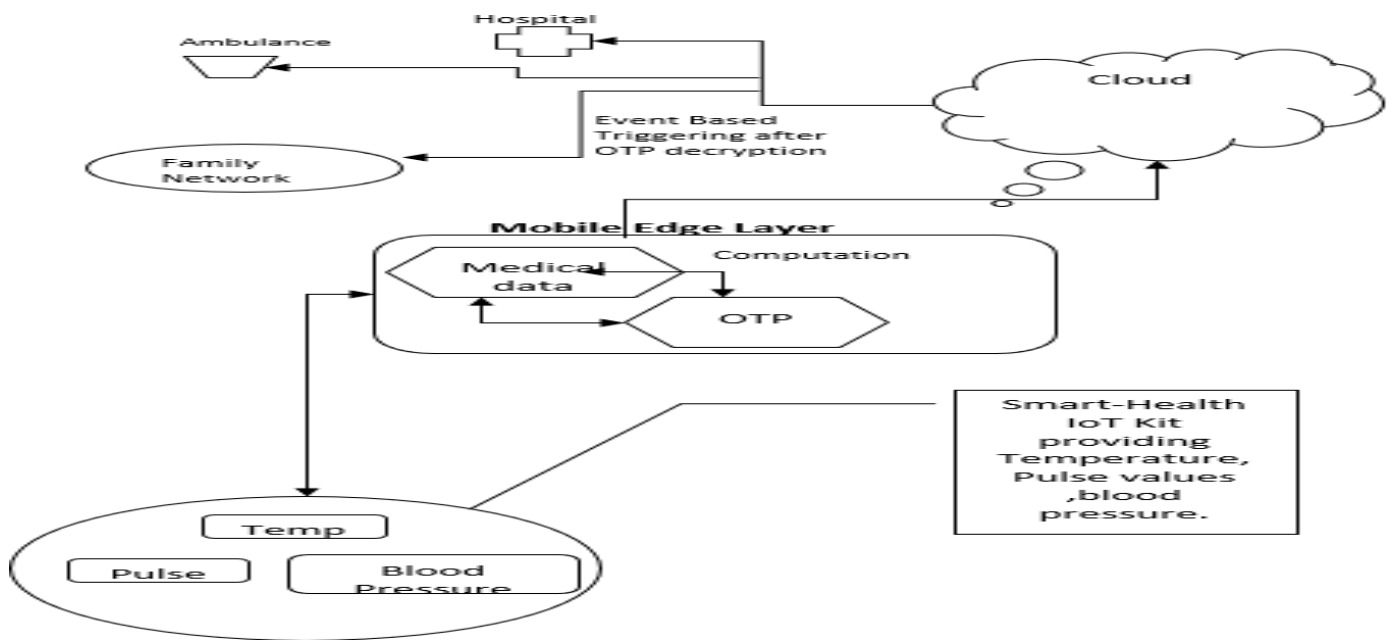


Figure 4 – Proposed System Architecture

SUMMARY AND CONCLUSION

Hence in this paper we have evaluated the sustainability and efficiency of edge computing using mobile networks. Specifically, we evaluated the performance of edge computing in health care application as an example system to understand the mobile edge concepts in depth. The end results have shown that edge computing is necessary for efficient less resource oriented and fail free system in health application and other cases. Although regional data centers

allow to significantly reduce network latency, only hosting resources at the edge enables a satisfactory quality of experience for proposed system. Furthermore, increasing the computational capabilities of the servers in the cloud does not compensate for the increase in network latency. Therefore, deploying even limited computing resources at the edge helps improve the quality of experience in the considered system. We hope that this article will encourage further research in this direction and evolve the edge computing technology further

REFERENCES

- [1] R. Li, H. Harai, and H. Asaeda, "An aggregatable namebased routing for energy-efficient data sharing in big data era," *IEEE Access*, vol. 3, pp. 955–966, 2015.
- [2] R. Li, H. Asaeda, and J. Li, "A distributed publisherdriven secure data sharing scheme for information-centric iot," *IEEE Internet of Things Journal*, 2017.
- [3] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the Internet of Things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, Apr.-June 2015.
- [4] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Int'l. Conf. Mobile Systems, Applications, and Services*, Bretton Woods, NH, June 2014.
- [5] S. Liszio and M. Masuch, "Designing shared virtual reality gaming experiences in local multi-platform games," in *International Conference on Entertainment Computing*. Springer, 2016, pp. 235–240.
- [6] J. Farman, "Locative life: Geocaching, mobile gaming, and embodiment," *Digital Arts and Culture* 2009, 2009.
- [7] S. Chess, "Augmented regionalism: Ingress as geomediated gaming narrative," *Information, Communication & Society*, vol. 17, no. 9, pp. 1105–1117, 2014.
- [8] E. Graells-Garrido, L. Ferres, and L. Bravo, "The effect of Pok'emon Go on the pulse of the city: A natural experiment," *arXiv preprint arXiv:1610.08098*, 2016.
- [9] ETSI, "Mobile-Edge Computing Introductory Technical White Paper," Sep. 2014.
- [10] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *ACM Proc. 1st Int'l. Workshop Mobile Cloud Computing & Networking (MobileCloud '13)*, Bangalore, India, July 2013.
- [11] H. Chang, A. Hari, S. Mukherjee, and T. Lakshman, "Bringing the cloud to the edge," in *IEEE Conf. Comp. Comm. Workshops (INFOCOM WKSHPS)*, April 2014.
- [12] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, 2017.
- [13] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, 2017.
- [14] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge architecture & orchestration," *IEEE Communications Surveys & Tutorials*, 2017.
- [15] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable crowd-sourcing of video from mobile

devices,” in Proc. of the 11th annual International conference on Mobile systems, applications, and services. ACM, 2013, pp. 139–152.

[16] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: research problems in data center networks,” ACM SIGCOMM computer communication review, vol. 39, no. 1, pp. 68–73, 2008.

[17] S. Wang, R. Uргаonkar, T. He, M. Zafer, K. L. Chan, and K. K. Leung, “Mobility-induced service migration in mobile micro-clouds,” in Military Communications Conference (MILCOM), 2014 IEEE. IEEE, 2014, pp. 835–840.

[18] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile fog: A programming model for large-scale applications on the internet of things,” in Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing. ACM, 2013, pp. 15–20.

[19] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1–1, 2016.

[20] L. Canzian and M. Van Der Schaar, “Real-time stream mining: online knowledge extraction using classifier networks,” IEEE Network, vol. 29, no. 5, pp. 10–16, 2015.

[21] Y. Elkhatib, B. Porter, H. B. Ribeiro, M. F. Zhani, J. Qadir, and E. Rivi`ere, “On using micro-clouds to deliver the fog,” IEEE Internet Computing, vol. 21, no. 2, pp. 8–15, 2017.

[22] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, “Federated edge-assisted mobile clouds for service provisioning in heterogeneous iot environments,” in Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on. IEEE, 2015, pp. 591–596.

[23] A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, “Where’s the bear?-automating wildlife image processing using iot and edge cloud systems,” in Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on. IEEE, 2017, pp. 247–258.

[24] ONF, “Software-defined networking: The new norm for networks,” ONF White Paper, 2012.

[25] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,” Mobile Networks and Applications, vol. 18, no. 1, pp. 129–140, 2013.

[26] Internet of Things for Smart Cities Andrea Zanella, Senior Member, IEEE, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, Senior Member, IEEE, and Michele Zorzi, Fellow, IEEE

[27] V.R.sarma Dhulipala ,Prathima Devadas and P.H.S Tejo murthy -Mobile phone Sensing Mechanism for Stress relax using Sensor Network. Wireless Personal Communication volume 86,page 1013-1022(201

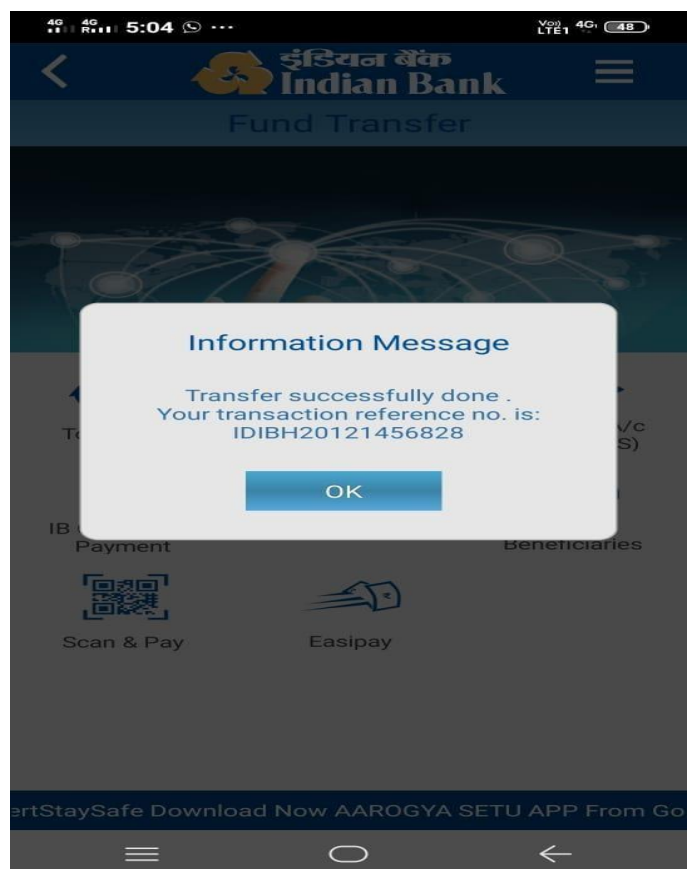
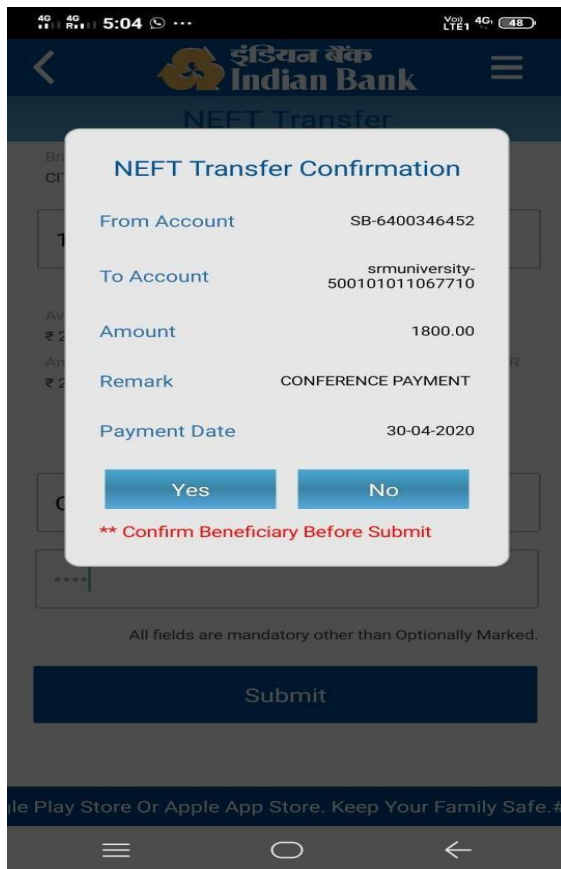
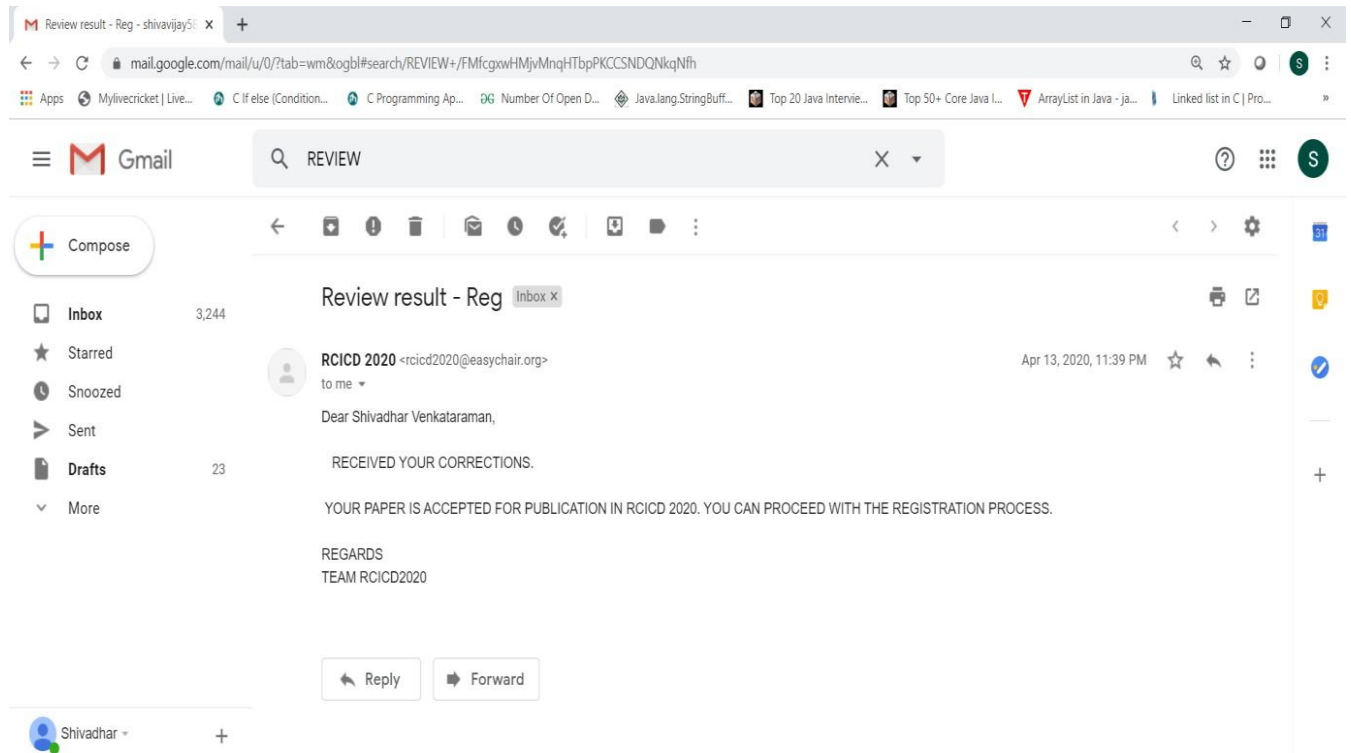
CONFERENCE DETAILS

APPLIED CONFERENCE NAME: RCICD2020 - RESEARCH CONFERENCE ON IOT , CLOUD AND DATA SCIENCE .

CONDUCTED BY: SRM UNIVERSITY

VENUE : SRM UNIVERSITY VADAPAZHANI CAMPUS.

PUBLICATION ACCEPTANCE AND PAYMENT PROOF



PLAGARISM REPORT

11
by 11 11

Submission date: 16-Mar-2020 10:50AM

Submission ID: 1274283262

File name: MOBILE_EDGE_COMPUTIONG_WITH_IOT_AND_ONE_TIME_PAD_ENCRYPTION.docx (355.12K)

Word count: 4962

Character count: 26751

MOBILE EDGE COMPUTING BASED IOT WITH ONE TIME PAD ENCRYPTION

ABSTRACT

We are in a period where cloud platform and edge computing are becoming a major platform for the computation, processing of various data and communication between machines, sensors and smart objects, in real time applications and for many crucial tasks. With IOT expanding in various real time applications various techniques of implementing IOT are emerging. It is important to understand the use of edge computing which will provide methodology for information sharing, data processing, data storing and data analyzing. The applications of Edge computing in IoT is keep on expanding with improving technology in day to day life. This Paper is to assist and help researchers working in the area of the Edge computing platform to understand concepts involved in edge computing easily. This paper also explains standard instructions to be followed on how to protect messages with one-time pad encryption easily in simplest way and how to implement mobile edge computing based IOT with least use of resources.

KEY WORDS: MOBILE EDGE COMPUTING, ONE TIME PAD ENCRYPTION, SECURITY

INTRODUCTION

Computers and electronic gadgets are required whenever rate of correctness, automation, speed and accurate computations are needed. In the conventional computing system, the input is given by human beings and the software applications running at the systems processes the inputted data and produces an accurate output as possible. Generally, the outputs are displayed on a output device such as a monitor or printer. As humans are lazy and there is possibility of inaccuracy in humans IOT systems are used. IOT Systems are 100 percent accurate and it eliminates possibility of human errors. The processing application is located at a central server in the case of an IoT system. The sensors in the IOT systems forward their data to a computing system via variety of networks such as mobile networks, Bluetooth, hotspot, WIFI, transmitter and receiver. The local computing system preprocesses all the data it obtains and sends it to a central server continuously using networks. In IOT an actuator or event-based notification system is used. Thus, an IoT system uses sensors, actuators and processors together to generate output. So IoT helps in improving automation in industries and cooperate companies. so we can summarize an IoT-enabled system is set of sensors, embedded system, electronics and communication devices which implements data storage via Internet into a cloud server and uses an analytics software to make decisions at the cloud platform uses help of analytics system to make certain decisions which is sent back to local device or end user. At periodic intervals, the collected data are analyzed using analytics software running in the server (cloud). The decisions and results from server are then informed accordingly to the end user. In most cases, it turns out to be the notification in the mobile device or actuation of some subsystems of the end device. We can use various technologies and concepts along with IOT. IOT can be implemented together with edge computing, fog computing, mobile computing and cloud computing or combination of these concepts. IN this paper we are going to see Different ways of implementing IOT with example system.

RELATED WORK

Wireless sensor networks consist of resource constrained devices that sense, process and transmit data from their surroundings to main server. They are launched in places that can be reached by humans. Due to their nature, many applications are available for their use such as health, military, monitoring and detecting abnormality in various cases[12]. Due to their computational speed, computational power and transmission constraints, the efficient use of sensors and resource constrained devices are essentially important. The mobile based communication, for example, places the highest strain on the energy conservation of the nodes [13]. Due to the energy consumption of data reception being twice as much as that of data processing in the Micas [14] mote, utilization of mobile communication is kept as short as possible. Ensuring that sensors work efficiently for a lengthy period comes with challenges [15] such as routing [16], data aggregation [17], clustering, and security. Security challenges include data security, physical compromise of nodes, and management and distribution of keys [18]. Data security challenges border around freshness and replay protection, integrity, authentication and confidentiality [7]. In satisfying data security concerns, existing protocols generate overheads related to time, energy and packet size. There have been surveys to determine which encryption algorithms will work best for encryption protocols. It has been observed that the Advanced Encryption Standard (AES) [19] is the most efficient, if done properly, when implemented in hardware [20]. Even though there are several encryption protocols, this paper mentions the key ones being Zigbee [23], C-Sec [21], Secure Network and Encryption Protocol (SNAP) [24], MiniSec [25] and TinySec [26].

EXISTING SYSTEM

IoT based remote monitoring systems are suggested by many professionals because of their reliability, time saving properties, accuracy and speed. In existing system, the IoT devices will keep observe the data from various resources and

send data to the respective fog nodes for computation of health data. Then the computed values will be transferred to the cloud Server. Based on the abnormal condition the Cloud will send intimation to the clients. Use of fog nodes is not that efficient as it makes the **network complex and inefficient**. This result in massive storage accumulation in cloud storage, processing the collected data is obviously a hard task to perform which results in computational overhead and delay in sending emergency alert.

PROPOSED SYSTEM

We propose the Secured OTP IOT Environment in Medical Field. Here the patient's android mobile device will be acting as the edge server for doing computing with the receiving data from health monitoring devices. The health-related data get uploaded into Mobile node. In Mobile Edge Layer, edge level computing is performed based on the threshold values the health status of respective patient is analyzed. If the status is getting abnormal then the Mobile Edge layer sends the health data with One-time pad security to the Cloud. Then the emergency SOS message will be sent immediately to ambulance, friends, close relatives, neighbors based on the type of event triggered. We propose in this system that the SMS is sent even without internet connectivity as it is for lifesaving and emergency purpose. Mobile computing promotes highly efficient output because maximum computation is done edge level which makes the cloud process as light weight. This system comes over the disadvantages of existing system and increase efficiency and decrease network and software requirements.

COMPARISON BETWEEN FOG COMPUTING, CLOUD COMPUTING AND MOBILE EDGE COMPUTING IN IOT.WHICH IS MORE EFFICIENT FOR PROPOSED SYSTEM?

IMPLEMENTATION USING CLOUD

Cloud provides data storage and hardware services virtually where the processing hardware is in random location. In cloud computing the data is sent to some **data processor located in virtual location** Where the data is processed and the result is sent back to the main server or the user application. This requires high speed internet and cloud service provider for cloud server which is time consuming and another problem is data maybe lost or easily hacked during tranctaction.For the proposed system high speed internet may not be available and it is required that in less time data is processed so implimenting using cloud is not much efficient .Availability is another problem with cloud. For the proposed system we need server available for 24/7 .In cloud we may have to send the data to randomly located long distance server in which there may be data loss during transfer and because of long distance data tranfer there are security issues .Data may be tampered with during transfer .So cloud model is not considered for the proposed system.

IMPLEMENTATION USING FOG COMPUTING

Fog is a kind of distributed networking model that is closely associated with cloud computing and the internet of things. In fog computing multiple fog nodes are created each nodes are given individual data and each node process data individually and the data are combined and result is generated. Use of fog nodes is not that efficient as it makes the **network complex and inefficient**. This result in massive storage accumulation in cloud storage, processing the collected data is obviously a hard task to perform which results in computational overhead and delay in sending emergency alert. The huge disadvantage with fog computing is it cannot handle huge quantity of data continuously .Implimentation and maintance of fog is complex.Comparing with requirements of our proposed system implimenting using fog will not be much efficient .

IMPLEMENTATION USING MOBILE EDGE COMPUTING

IoT Mobile Edge Computing is a combination of tiny data centers locally placed where most of the data is stored and processed locally. These network of tiny data centers are connected in the form of tree, mesh, point to point, star topologies and they push the data received to a centrally located storage repository. The range supported is **within a range of 10-meter square** or less. So basically, Edge Computing is used for data analysis, storing and **processing of data which are closer to the sensors that provide the data**. Smart devices and sensors used in IoT Edge Computing are capable of processing critical data fragments and provide a quick real time response. These devices and sensors prevent the delay caused by sending the data through internet to cloud and back from cloud to local system. These devices also don't require high speed network. These devices are designed to act as small data centers that provides no latency. With this enhanced capability of Mobile edge computing, data processing is made to be implemented locally in decentralized server which effectively reduces the network traffic. This data can later be collected by the main server for further evaluation and processing. The data from the sensors or devices are sent to MEC Gateway (Mobile edge computing gateway) where the data is processed and analyzed to produce results. Mobile edge computing enhances the Cloud computing capabilities close to the Radio Access mobile Networks **4G, 5G, LTE, CDMA, Bluetooth, etc** is developing a system architecture and standard for a few Application program interface. Mobile edge computing manages local cache data local data breakout data analytics aggression. Virtualization in edge computing enables cloud computing providers to execute many independent application instances in a single physical server. In edge computing all the data is processed locally which is easier to manage. the user can track data and it is much secure and easy. This kind of architecture reduces complexity of implementation. The local device which process data can be made available 24/7 according to the usage of user .so we have considered to impliment the system using this concept.

Approach	Edge nodes	Edge network
cloudlets	Compact size datacenters deployed on WiFi access	WiFi,3G,LTE
Mobile cloudlets	Compact size datacenters on cars	3G, LTE
Multi access edge computing	Servers deployed in radio servers	3G,LTE,Other access technologies
Fog computing	Heterogeneous nodes including high end routers, access point setup box	Multiple wireless access technologies including WiFi,3G,LTE
Mobile cloud	Neighboring mobile nodes form a cloud with one device chosen as resource coordinator	Local networking through WiFi Bluetooth internet connectivity with wifi,3G,LTE
Edge cloud	Compute or storage nodes deployed in edge network and federated to cloud data centers	Home/enterprise networks and WiFi hotspots
Fusion	Service nodes deployed on access points, local and centralized data centers	Not defined

Table 1- Different approaches in Edge

CLASSIFICATION OF EDGE COMPUTING PLATFORMS

There are three classifications (A) resource rich servers that are deployed at the edge level , (b) heterogeneous edge based computing nodes and (c) edge to cloud federation. The classification is based on how edge is deployed. However, in practice, one category can be used by combining one category with others categories. One category relies on servers with rich set of resources that are deployed very close(below 100 feet) to the end-devices. Another group uses the resources from various nodes present at the edge, including the end devices themselves. A third category is based on the federation of resources at the edge nodes and centralized data centers. each category is explained below in detail

A)RESOURCE RICH SERVERS DEPLOYED AT EDGE:

One option to realize an edge computing platform is to deploy the resource-rich servers in the network to which end-users are connected. A cloudlet is defined as a set of data centers that offers sets of clusters of multicore computing units, data storage and wireless LAN connectivity towards the edge. All the sensor data collected from sensors are processed on the cloudlet to provide real-time assistance and analysis assistance. Since the cloudlets are launched and introduced, further research professionals have proposed integration of cloudlets with femtocells, the LTE base stations define the architecture of micro data centers consisting of thousands of servers which are capable of hosting interactive applications for end-users.

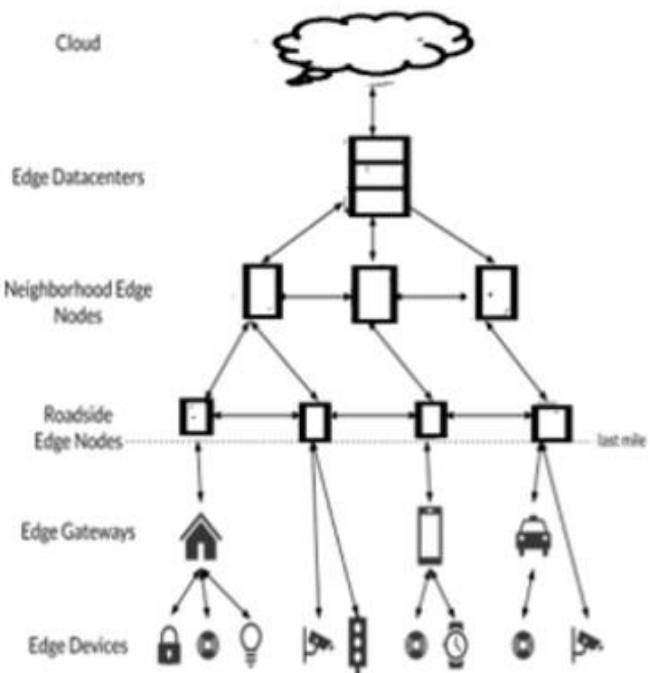


Figure 1-Edge computing architecture

While these data centers have been used for deploying Content Delivery networks(CDNs) and email applications ,they can also be reengineered to host cloudlets Similar to cloudlets which will be deploying a small set of servers on WiFi access points or a base station in the radio access network is called as a micro cloud.

B)HETEROGENEOUS EDGE NODES:

In contrast to the set of techniques and solutions that are described above, edge computing platforms can produce a diverse set of computing resources. It consists of a fog platform devices (including smartphones and connected vehicles). Heterogeneity of wireless connectivity is a key aspect of the end-devices. Thus, many kinds of wireless access network(WAN) technologies are supported by their solution. fog computing is a cooperation based mobile cloud, where in heterogeneous mobile devices opportunistically share their resources to deliver particular and appropriate services and applications which are characterized by a highly virtualized system of heterogeneous nodes, ranging from resource-rich servers to more constrained edge routers, access points, set-top boxes and even end .

C) EDGE CLOUD FEDERATION MODEL :

Another option to understand edge platforms is based on the combination of computing resources at the edge and centralized data centers describe this concept as an edge cloud.

In this system, edge applications are used to deliver services and resources at the edge as well as in distant cloud centers. Many professionals advice the use of edge apps to launch indoor 3D localization and video monitoring applications. Similarly, the correct combination of private and public clouds enable integrated IoT applications. In this Type of edge

computing system, the edge node continuously orchestrates the federation of nodes to maximize the number of tasks that are executed. For the fusion architecture in this model federation of cloud is important factor. In this edge model, the services are deployed on a cloud infrastructure which are then distributed throughout the Internet.



Figure 2 classification of edge models A)Resource rich edge server B)Heterogeneous edge nodes C)Edge-cloud federation

MONITORING HEALTH DATA USING EDGE BASED IOT WITH OTP

This project is to explain the practical implimentation of mobile edge computing based iot and otp encryption. We can split the project into 4 modules

1. User Authentication in Web-portal
2. Patient's Mobile Application and Appointment
3. Mobile Computation
4. Event Based Triggering

MODULE 1 USER AUTENTICATION IN WEB PORTAL:-

In this module we have developed a web portal – Hospital Web application. Using this application new patients has to register their details and the data will be stored in hospital Server. Likewise a hospital Admin sign-in will be there they can add new doctors and specific specialist to which the doctors belong to. All these informations will be saved into hospital server's database. We can use jsp to create web application easily. Java is a kind of programming language which has one of its major characteristic as a complete independence from the physical platform on the level of translated code. Java applet and java servlet page **USP** can be used to develop Web pages distribute them to users. Using jsp we combine major java concppts such as jdbc, seing, hibernate to create web server. It is recommended to code in advanded ide such as eclipse and netbeans. Because of the given features and **reliability of java web development** we implement the system using java.

MODULE 2 PATIENT'S MOBILE APPLICATION

In this module we create mobile application for patient this application monitors patient's health continuously and it maintains patient's health history. The patient communicates with this system by first registering his/her information at first instance by answering few questions related to health history and personal details. After registration is successfully done, a unique number for identification purpose is provided to the patient by the cloud server. To differentiate users, cloud layer provides identification such as patient identification (PID) and the Server also provides sets of attributes related to health history of the patients. This module works in event triggered mode. In this mode, the requisite real-time sampled data is stored at the mobile nodes. The mobile edge layer will conduct a data handling. Patients can request appointment to their respective Doctors from mobile application or web application. The mobile application can be easily developed for android using android studio using java concepts.

MODULE 3 MOBIL EDGE COMPUTATION

The health data from various sensors from respective areas will be processed in Mobile layer. We use edge-federation model of edge computing which is found to be efficient for this project implimentation. Patient's own mobile device acts as the main resource for computation. Based on the previous health dataset the computation will be processed. Data from different categories will be analyzed here. Health related data from previous history got collected from the Health dataset, Environment related data like air quality, noise level around the place where patient is, Behavior related data like whether the patient is having fits, vomiting, hyper tension, fainting etc. These kinds of data get analyzed in this layer. All these data will be computed in the users mobile phone itself. The patient will

be given watch like device will consist of set of sensors these sensors will send the data to the patient's mobile application through mobile technology such as hotspot,bluetooth.These data are then computed in patients phone itself.All the definitions and algorithms for the computations are defined in The patient's mobile application itself .The patient's mobile app generates a result and based on the result the application will conclude status of the patient based on critical values defined .The entire computation is controlled by the patient's mobile application .Hardware resources of the patients phone are primarily used for computation . After computing, the end result in form of report which is in text format will be sent to the one-time-pad security mechanism.The one time pad encryption and decryption will be implemented as given below.

ONE TIME PAD ENCRYPTION

One-time pad encryption known as most secure and stable method to protect short text messages. This paper explains how to implement one-time pads, and how to deal with its various security issues. Working with one-time pads is easy to learn. The one-time -pad system is transparent and we dont need any computer, special equipment or any knowledge about

cryptography techniques or mathematics .to impliment one time pad encryption . The system will surely provides unbreakable encryption when properly usedand implimented. It won't be possible to crack one-time pad encrypted data without having the proper key, even as future attack technology may evolve .The one-time pad (OTP), also called as Vernam cipher or the perfect cipher, is a cryptographic algorithm where plain text is combined with a random key to produce an encrypted data. The OTP is uncrackable only if four important rules are followed. When rules are applied accurately, the one-time pad can be proven unbreakable. However, if any one of the given rules is not satisfied, the cipher is no longer unbreakable. The following

1. The minimum length of key is as long as the plaintext.
2. The technique of key generation is kept between the users.
3. There must only be 2 copies of the key: one key for the sending user and another for the receiving user (some exceptions exist for multiple receivers).
4. The keys are to be used only once, and both sending user and receiving must destroy each of their keys after use

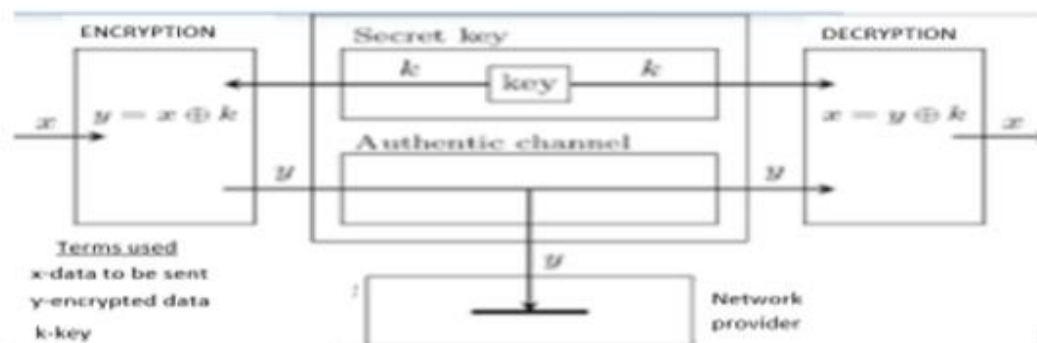


Figure 3-One time pad diagrammatic representation

IMPLEMENTING ONE TIME PAD ENCRYPTION

To perform one-time pad encryption we need a key, called one-time -pad- key.The key is generated based on random technique users choose . One-time pad encryption is only possible if both sender and receiver are in possession of the same key which needs to be communicated or technique of key generation has to be known by the users. Therefore, both sender and receiver must exchange their keys beforehand and be ready. one key could be sufficient for a very long period as long as it is kept secret by users.

ENCRYPTION

To perform encryption First covert all data to hexadecimal and store it in variable.we get the key from length of data,perform

XOR operation between the data and key to get the encrypted data. Now the data is completely secure ciphertext! Let us see an example with medical data

Bp 110

Pulse 65

1)hexadecimal form of above data

4270313130

50756c7365203635

2)key(length=26)

3236

4)hexadecimal to text

Sp 100

Pulse 65

After decryption the receiving user can successfully view the message .

We create the encryption in the application itself . The application automatically encrypts the data .We are create the encryption algorithm using arithmetic and bitwise operators in java concepts.The

MODULE 4 EVENT BASED TRIGGERING

This is Stage where alert SMS is sent to patient's known members and ambulance. The members to whom the SMS containing patient's health report is to be sent is decided based on the result of the patient's status. The status and health report sent will be in encrypted format. When the mobile device performs computation periodically when the values got reached above the threshold value then it will detect that patient is in emergency state and has abnormal condition. So spontaneously the mobile node will send the abnormal data to the Cloud in encrypted format and here the Event is triggered by the cloud server. The Emergency alert to the Doctors, medical team, ambulance, relatives, etc. Thus the alert will be sent to respective mobile users. Set of actions are done according to the result from edge computation. At the receiver end the user has to click the received text and enter the key to obtain the original text. The main task of this module is to choose the members whom are supposed to receive the SMS. This module chooses the members that should receive the message based on the result.

4270310306

2)key calculation

Length=26=3236

```
3)decryption(reverse xor)
```

4270313130

50756c7365203635



SUMMARY AND CONCLUSION

Hence in this paper we have evaluated the sustainability and efficiency of edge computing using mobile networks. Specifically, we evaluated the performance of edge computing in health care application as an example system to understand the mobile edge concepts in depth. The end results have shown that edge computing is necessary for efficient less resource oriented and fail free system in health application and other cases. Although regional data centers

allow to significantly reduce network latency, only hosting resources at the edge enables a satisfactory quality of experience for proposed system. Furthermore, increasing the computational capabilities of the servers in the cloud does not compensate for the increase in network latency. Therefore, deploying even limited computing resources at the edge helps improve the quality of experience in the considered system. We hope that this article will encourage further research in this direction and evolve the edge computing technology further.

REFERENCES

- [1] R. Li, H. Harai, and H. Asaeda, "An aggregatable namebased routing for energy-efficient data sharing in big data era," *IEEE Access*, vol. 3, pp. 955–966, 2015.
- [2] R. Li, H. Asaeda, and J. Li, "A distributed publisher-driven secure data sharing scheme for information-centric iot," *IEEE Internet of Things Journal*, 2017.
- [3] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the Internet of Things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, Apr.-June 2015.
- [4] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Int'l. Conf. Mobile Systems, Applications, and Services*, Bretton Woods, NH, June 2014.
- [5] S. Liszio and M. Masuch, "Designing shared virtual reality gaming experiences in local multi-platform games," in *International Conference on Entertainment Computing*, Springer, 2016, pp. 235–240.
- [6] J. Farman, "Locative life: Geocaching, mobile gaming, and embodiment," *Digital Arts and Culture* 2009, 2009.
- [7] S. Chess, "Augmented regionalism: Ingress as geomediated gaming narrative," *Information, Communication & Society*, vol. 17, no. 9, pp. 1105–1117, 2014.
- [8] E. Graells-Garrido, L. Ferres, and L. Bravo, "The effect of Pok'emon Go on the pulse of the city: A natural experiment," *arXiv preprint arXiv:1610.08098*, 2016.
- [9] ETSI, "Mobile-Edge Computing Introductory Technical White Paper," Sep. 2014.
- [10] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *ACM Proc. 1st Int'l. Workshop Mobile Cloud Computing & Networking (MobileCloud '13)*, Bangalore, India, July 2013.
- [11] H. Chang, A. Hari, S. Mukherjee, and T. Lakshman, "Bringing the cloud to the edge," in *IEEE Conf. Comp. Comm. Workshops (INFOCOM WKSHPs)*, April 2014.
- [12] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, 2017.
- [13] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, 2017.
- [14] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge architecture & orchestration," *IEEE Communications Surveys & Tutorials*, 2017.
- [15] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable crowd-sourcing of video from mobile devices," in *Proc. of the 11th annual international conference on Mobile systems, applications, and services*, ACM, 2013, pp. 139–152.
- [16] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68–73, 2008.
- [17] S. Wang, R. Ugaonkar, T. He, M. Zafer, K. L. Chan, and K. K. Leung, "Mobility-induced service migration in mobile micro-clouds," in *Military Communications Conference (MILCOM)*, 2014 IEEE, 2014, pp. 835–840.

- [18] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenw"alder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, ACM, 2013, pp. 15–20.
- [19] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2016.
- [20] L. Canzian and M. Van Der Schaar, "Real-time stream mining: online knowledge extraction using classifier networks," *IEEE Network*, vol. 29, no. 5, pp. 10–16, 2015.
- [21] Y. Elkhatib, B. Porter, H. B. Ribeiro, M. F. Zhani, J. Qadir, and E. Riviere, "On using micro-clouds to deliver the fog," *IEEE Internet Computing*, vol. 21, no. 2, pp. 8–15, 2017.
- [22] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "Federated edge-assisted mobile clouds for service provisioning in heterogeneous iot environments," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, IEEE, 2015, pp. 591–596.
- [23] A. R. Elias, N. Golubovic, C. Krintz, and R. Wolski, "Where's the bear?-automating wildlife image processing using iot and edge cloud systems," in *Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on*, IEEE, 2017, pp. 247–258.
- [24] T. Taleb, A. Ksentini, and R. Jantti, "Anything as a Service for 5G mobile systems," *IEEE Network*, vol. PP, no. 99, pp. 12–19, Nov 2016.
- [25] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and linux containers," *IBM Research Report*, July 2014.
- [26] ETSI, "Network Functions Virtualisation - An Introduction, Benefits, Enablers, Challenges, Call for Action," *Tech. Rep.*, Oct. 2012.
- [27] ONF, "Software-defined networking: The new norm for networks," *ONF White Paper*, 2012.
- [8] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.
- [39] Y. Lin, T. K"am" ar"ainen, M. Di Francesco, and A. Yi" au" a"aski, "Performance evaluation of remote display access for mobile cloud computing," *Computer Communications*, vol. 72, pp. 17–25, December 2015.
- [30] Internet of Things for Smart Cities Andrea Zanella, Senior Member, IEEE, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, Senior Member, IEEE, and Michele Zorzi, Fellow, IEEE

10

ORIGINALITY REPORT

14%

SIMILARITY INDEX

7%

INTERNET SOURCES

7%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

users.aalto.fi

Internet Source

7%

2

Kwasi Boakye-Boateng, Eric Kuada, Emmanuel Antwi-Boasiako, Emmanuel Djaba. "Encryption Protocol for Resource-Constrained Devices in Fog-Based IoT Using One-Time Pads", IEEE Internet of Things Journal, 2019

Publication

7%

Exclude quotes On

Exclude bibliography On

Exclude matches Off