A PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE INTERNET OF THINGS BASED ON BLOCK CHAIN

A project report submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In partial fulfillment of the requirements for the awarded of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

B CHAITANYA KUMAR REDDY	**(20AP1A0507)**
M VAISHNAVI DEVI	**(20AP1A0534)**
K PRASANNA LAKSHMI	**(20AP1A0528)**
K SAI DURGA PRASAD	**(20AP1A0530)**

Under the esteemed guidance of

Mr. G S V R ABHISHEK

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BHIMAVARAM INSTITUTE OF ENGINEERING AND TECHNOGOLY

Affiliated to JNTU, KAKINADA and Approved by AICTE, NEW DELHI

PENNADA, BHIMAVARAM-534243

(2020-2024)

BHIMAVARAM INSTITUTE OF ENGINEERING AND TECHNOLOGY

Affiliated to JNTU, KAKINADA and Approved by AICTE, New Delhi

Pennada, Bhimavaram-534243.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certificated by the project work entitled **“A PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE INTERNET OF THINGS BASED ON BLOCK CHAIN”** is the bonafied work done by ,B CHAITANYA KUMAR REDDY(20AP1A0507),M VAISHNAVI DEVI (20AP1A0534),K PRASANNA LAKSHMI(20AP1A0528), K SAI DUGRA PRASAD(20AP1A0530), in the department of **COMPUTER SCIENCE AND ENGINEERING** during the academic year **2020-2024.** This work has been carried out under my guidance and super vision the result embodied in this project report have not been submitted in any university of organization for the award of any degree (or) diploma.

Internal Guide	**Head of the Department**
Mr. G S V R ABHISHEK	**Mr. U S V VINOD**
Assistant Professor	Associate Professor
Department of CSE	Department of CSE
Internal Examiner	External Examiner

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon me their blessings and the heart pledged support, this time I am utilizing to thank all the people who have been concerned with this project.

We would like to thank my guide **Mr. G S V R ABHISHEK**, whose valuable guidance has been the ones that helped me to patch this project and make it full proof success.

We would like to thank my principal **Dr. K. SURESH sir, Bhimavaram Institute of Engineering and Technology,Pennada , Bhimavarm** whose encouragement and guidance helped me for the fulfillment of the project.

Then We would like to thank my Head of the Department CSE **Mr. U S V VINOD,** who have helped me with their valuable suggestions and guidance has been very helpful in various phases of the completion of the project.

Last but not the least I would like to thank my classmates who have helped me a lot.

By:

B CHAITANYA KUMAR REDDY	(20AP1A0507)
M VAISHNAVI DEVI	(20AP1A0534
K PRASANNA LAKSHMI	(20AP1A0528)
K SAI DURGA PRASAD	(20AP1A0530)

DECLARATION

We here by declare that project report entitled **“A PROXY RE-ENCRYPTION APPROACH TO SECURE DATA SHARING IN THE INTERNET OF THINGS BASED ON BLOCK CHAIN”** is genuine project work carried out by us in Bachelor of Degree **COMPUTER SICENCE AND ENGINEERING** Bachelor course of **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** and has not been submitted to any other courses (or) university foraward of degree of us.

B CHAITANYA KUMAR REDDY	(20AP1A0507)
M VAISHNAVI DEVI	(20AP1A0534)
K PRASANNA LAKSHMI	(20AP1A0528)
K SAI DURGA PRASAD	(20AP1A0530)

Date:

Place:

CONTENTS

LIST OF FIGURES

ABSTRACT

The evolution of the Internet of Things has seen data sharing as one of its most useful applications in cloud computing. As eye-catching as this technology has been, data security remains one of the obstacles it faces since the wrongful use of data leads to several damages. In this article, we propose a proxy re-encryption approach to secure data sharing in cloud environments. Data owners can outsource their encrypted data to the cloud using identity-based encryption, while proxy re-encryption construction will grant legitimate users access to the data. With the Internet of Things devices being resource-constrained, an edge device acts as a proxy server to handle intensive computations. Also, we make use of the features of information-eccentric networking to deliver cached content in the proxy effectively, thus improving the quality of service and making good use of the network bandwidth. Further, our system model is based on block chain, a disruptive technology that enables decentralization in data sharing. It mitigates the bottlenecks in centralized systems and achieves fine-grained access control to data. The security analysis and evaluation of our scheme show the promise of our approach in ensuring data confidentiality, integrity, and security.

1. INTRODUCTION

1.1 INTRODUCTION

THE Internet of Things (IOT) has emerged as a technology that has great significance to the world nowadays and its utilization has given rise to an expanded growth in network traffic volumes over the years. It is expected that a lot of devices will get connected in the years ahead. Data is a central notion to the IOT paradigm as the data collected serves several purposes in applications such as healthcare, vehicular networks, smart cities, industries, and manufacturing, among others [1]. The sensors measure a host of parameters that are very useful for stakeholders involved. Consequently, as enticing as IOT seems to be, its advancement has introduced new challenges to security and privacy. IOT needs to be secured against attacks that hinder it from providing the required services, in addition to those that pose threats to the confidentiality, integrity, and privacy of data.

A viable solution is to encrypt the data before outsourcing to the cloud servers. Attackers can only see the data in its encrypted form when traditional security measures fail. In data sharing, any information must be encrypted from the source and only decrypt by authorized users in order to preserve its protection. Conventional encryption techniques can be used, where the decryption key is shared among all the data users designated by the data owner. The use of symmetric encryption implies that the same key is shared between the data owner and users, or at least the participants agree on a key. This solution is very inefficient. Furthermore, the data owners do not know in advance who the intended data users are, and, therefore, the encrypted data needs to be decrypt and subsequently encrypted with a key known to both the data owner and the users. This decrypt-and-encrypt solution means the data owner has to be online all the time, which is practically not feasible. The problem becomes increasingly complex when there are multiple pieces of data and diverse data owners and users.

1.1.1 DATA ENCRYPTION AND PROXY RE-ENCRYPTION (PRE)

Although simple, the traditional encryption schemes involve complex key management protocols and, hence, are not apt for data sharing. Proxy re-encryption (PRE), a notion first proposed by Blaze , allows a proxy to transform a file computed under a delegation's public

1.1.2 COMBINING PRE WITH IBE, ICN, AND BLOCKCHAIN

Motivated by this scenario, this article proposes an improvement in IOT data sharing by combining PRE with identity based encryption (IBE), information-centric networking (ICN), and block chain technology. Shamir first presented the notion of IBE, in which a sender encrypts a message to a recipient using the identity (email) as the public key. It is a very powerful primitive used to combat numerous key distribution problems and has consented to the development of several cryptography protocols, including public-key searchable encryption , secret handshakes , and chosen cipher text attack (CCA) secure public-key encryption . IBE is preferred over attribute-based encryption (ABE) because ABE involves heavy computations on data encryption, decryption, and key management, and these processes are not convenient for the resource-constrained IOT devices. The strength of this article is increased by borrowing the idea of ICN to cater for the growth in information sharing.

revocation can also be achieved using block chain. PRE, together with IBE and the features of ICN and block chain, will enhance security and privacy in data-sharing systems.

1.2 Project Overview

This project focuses on enhancing IOT data sharing by combining Proxy Re-Encryption (PRE) with Identity-Based Encryption (IBE), Information-Centric Networking (ICN), and Block chain technology. These technologies are integrated to address the challenges of security, privacy, and data access control in IOT environments. The project aims to provide a secure and efficient framework for data sharing among IOT devices while ensuring confidentiality, integrity, and privacy of the data.

1.3 Objective

PRE and IBE will ensure fine-grained data access control, while the concept of ICN promises a sufficient quality of service in data delivery because the in-network caching provides efficient distribution of data. The block chain is optimized to prevent storage and data-sharing overheads and also to ensure a trusted system among entities on the network. In our article, the data owner propagates an access control list which is stored on the block chain. Only the authorized users are able to access the data. The contributions of this article are summarized as follows.

1) We propose a secure access control framework to realize data confidentiality, and fine grained access to data are achieved. This will also guarantee data owners’ complete control over their data

2. LITERATURE SURVEY

2.1 PRE Data Sharing

combined key-policy ABE (KP-ABE) and PRE to propose a system for data sharing in the cloud. The data was encrypted using KP-ABE which meant that only an appropriate collection of the attribute secret keys can make decryption possible. Besides the encrypted data, the cloud also managed all attribute secret keys except one special secret key in order to handle revocation of users. When users are revoked, new keys were distributed to the remaining users by the data owner and the encrypted data had to be re-encrypted. Although the scheme was efficient, the re-encryption was performed in a lazy way, and, therefore, the security of the scheme was weakened. Park provided a modification to the scheme in , where collusion between the service provider and revoked users is avoided. Their scheme was to basically replace the service provider with a trusted third party, which implies that there should be reliance on stronger trust assumption. Other schemes have made similar approaches but utilized cipher text-policy ABE (CP-ABE) rather, in which the access policy is associated with the cipher text instead of the secret keys. Liu et al. also proposed a time-constrained access control scheme based on PRE and ABE. ABE was used to design time-based access control policies while PRE was used to update the time attributes. Although these schemes have their advantages, they are not suitable in the context of IOT due to the heavy computations on encryption and decryption.

An IBE PRE scheme suitable for data sharing was presented by Han et al. in [21]. The re-encryption keys were not only bound to the users' identities but also to a specific cipher text. This implied that the data owner had to create a different re encryption key for each pair of data user and shared file. A similar idea was proposed by Lin et al. [22] where they used a hierarchical PRE instead of an identity-based PRE. These two schemes tend to be inefficient when multiple and complex data pieces are considered. An identity-based broadcast encryption (IBBE) combined with PRE was proposed by Zhou et al. in [23] for data sharing. Their scheme was a hybrid one that allowed the conversion to be done between the two protocols without leaking any sensitive information. Wang et al. [24] also designed an identity-based PRE (IBPRE) scheme for accessing health records. The scheme achieved coarse-grained access control. If a proxy receives the re-encryption key from the data owner, either all the cipher texts can be re-encrypted and accessible to the intended users or none at all. On that note, Shao et al. [25] proposed an IBE PRE scheme that is based on conditions. In their proposal, the proxy could transform a subset of cipher texts under an identity to other cipher texts under another identity. However, decryption rights to a group of users could not be authorized. In addition to the above, PRE has been used to mitigate security problems in IOT [26].

2.2 Block chain-Based Access Control and Data Sharing

Zyskind et al. [27] used block chain to provide distributed personal data management and ensure privacy as well. The block chain was utilized as an automatic access control manager, and, hence, no third party was required. Only the data address was stored on the block chain and a distributed hash table was used as the implementation of the data storage. This reduced the risk of data leakage. However, no specific access control model was proposed in their scheme. Maesa [28] proposed a block chain-based access control scheme where the data owner defines policies on the data and stores them on the block chain. The policies are then assigned to the users as access rights.

Fan et al. [29] designed a similar model to [28] where the encrypted data is uploaded to the cloud and access policies on the data are stored on the block chain as transactions. Although these two schemes achieve tamper-proof systems and easy auditing, there is a leakage of access policies since the block chains used are public ones and are thus visible to everyone. Singh and Kim [30] presented a block chain-based model for sharing data in vehicular networks and also enable secure communication among vehicles. However, the use of a public block chain does not work well in peer-to-peer (P2P) data sharing among vehicles due to the high cost involved in establishing a public block chain in resource-constrained vehicles.

2.3 Access Control Schemes for ICN

In order to control content in ICN frameworks, several centralized and decentralized access control mechanisms have been proposed in literature. Silva and Zorzo [31] presented an access control system for named data networking which relied on an ABE scheme and a proxy server. Before a content is published, the data owner encrypts the content and generates an access policy that binds it. The encrypted data is stored in the immediate routers while the access policy is stored on the server. When a user wants to access content, the user retrieves the content from the router, obtains the access policy from the proxy server, and then decrypt the data. Their scheme enables user revocation; however, it suffers from a single point of failure if a proxy server fails to work because the proxy server takes part in each content access. Li et al. [32] designed a privacy enhancing scheme using ABE for access control in ICN, and a trusted third party is deployed to manage attributes. A content publisher generates an access policy based on the attributes defined by the third party and uses a random symmetric key to encrypt the data. The publisher then hides the random key and the access policy in the content name and only authorized users can gain access to the content. The proposed scheme achieves privacy by hiding the access policy in the content name, but user revocation is not guaranteed.

For decentralized access control systems,Misra et al.[33] proposed a secure content delivery ICN framework using Shamir's threshold secret sharing scheme and broadcast encryption but without the services of a third party. A symmetric key is used to encrypt the content which is broadcast to the network along with the key generation materials. Only authorized users can use these keying materials and decrypt the encrypted data using their individual keys. The scheme provides user revocation services, but an account of each content access or the history of keying materials' update is not kept. This makes auditing difficult. Abdallah et al. [34] made use of the Diffie–Hellman (DH) protocol in the process of content publishing to achieve decentralized access control. The content, its name, and metadata are sent to the ICN, while only the content name is published. After going through the various stages of the DH key exchange protocol, the ICN verifies the metadata and sends the encrypted data together with the shared key. There is no single point of failure in this scheme; however, the cached content in the ICN is in the plain text form which makes it vulnerable to attacks.

Cloud servers are used to facilitate IOT data sharing and provide seamless, efficient, and robust sharing services in [35]– [37]. However, there are privacy concerns [38], [39]; the cloud is not trusted, and, hence, it is indispensable to enforce data access control over potentially un trusted platforms. Besides these, several schemes [40]–[42] are based on ABE. Although they are efficient, the high computations in key generation and distribution are not opportune for IOT. Inspired by the drawbacks in the applications of the various technologies for access control and data sharing, this article utilizes PRE, IBE, and the features of ICN and block chain to solve the challenges in data sharing. To the best of our knowledge, this article is the first to combine these mechanisms to establish secure data sharing in the cloud. Ateniese et al. [43] proposed a re-encryption scheme that is unidirectional, non interactive, of multi use, and non transitive. These properties are suitable for our proposed architecture, and, hence, the scheme is adopted in this article. A detailed construction of the security proof is also provided.

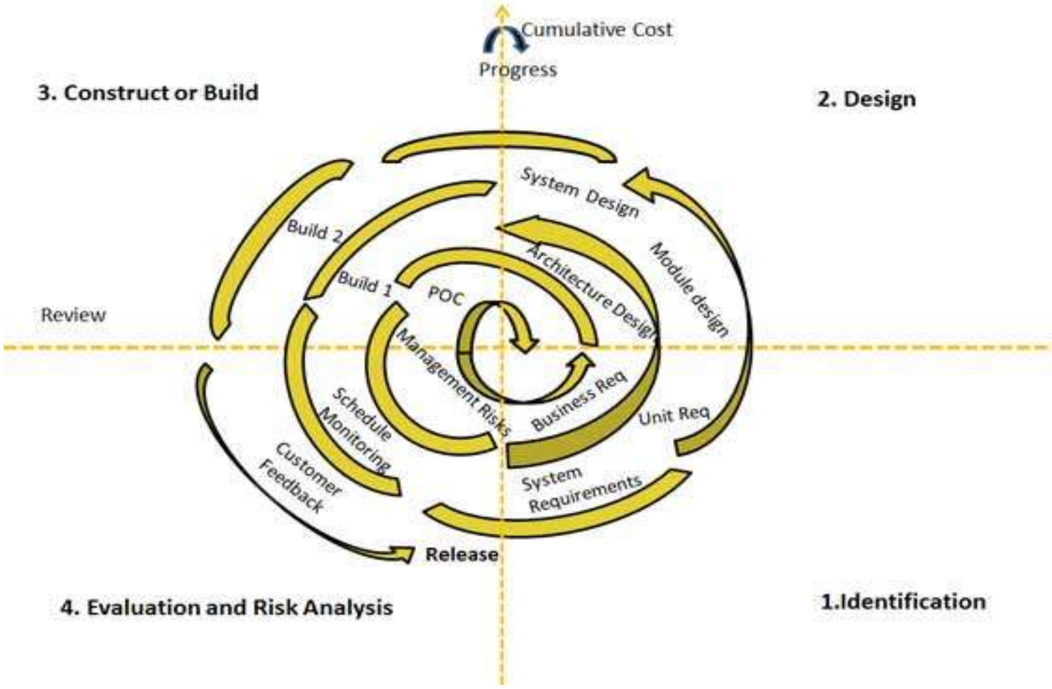
3. SYSTEM ANALYSIS

3.1 INTRODUCTION

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

3.2 SDLC (Spiral Model):



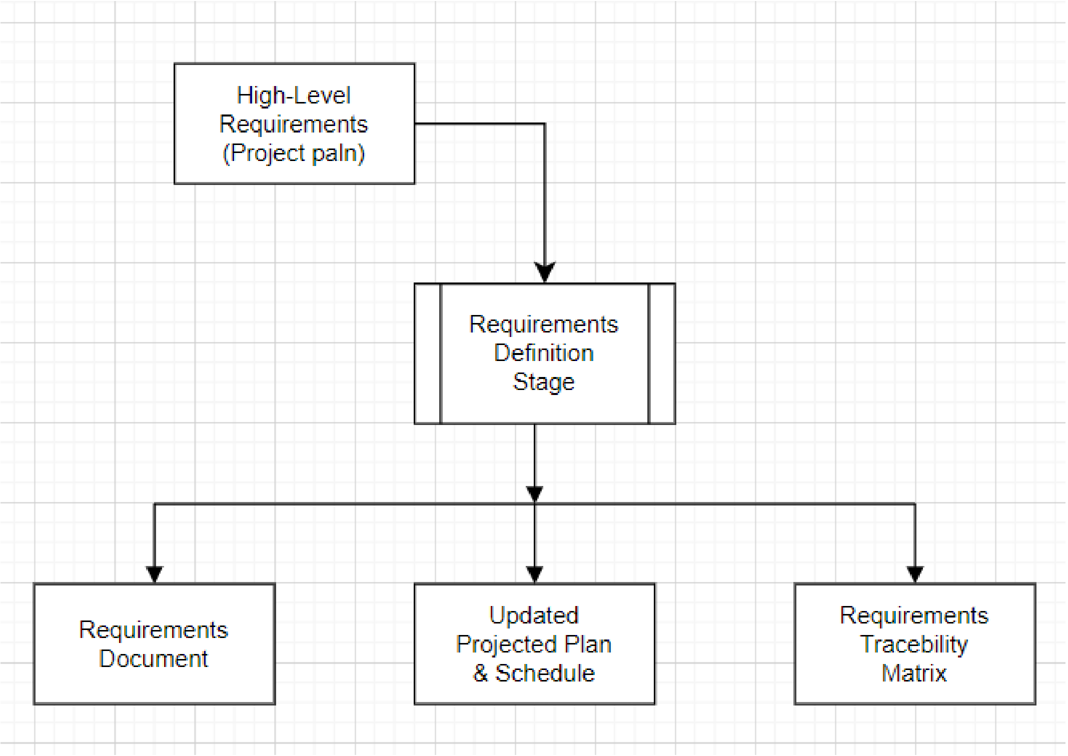
3.2.1 Stages of SDLC:

Requirement Gathering and Analysis

- Designing
- Coding
- Testing
- Deployment

Requirements Definition Stage and Analysis:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.


High-Level Requirements (Project paln)
Requirements Definition Stage
Requirements Document
Updated Projected Plan & Schedule
Requirements Tracebility Matrix

These requirements are fully described in the primary deliverable for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). the requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development life cycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*. The outputs of the requirements definition stage include the requirements

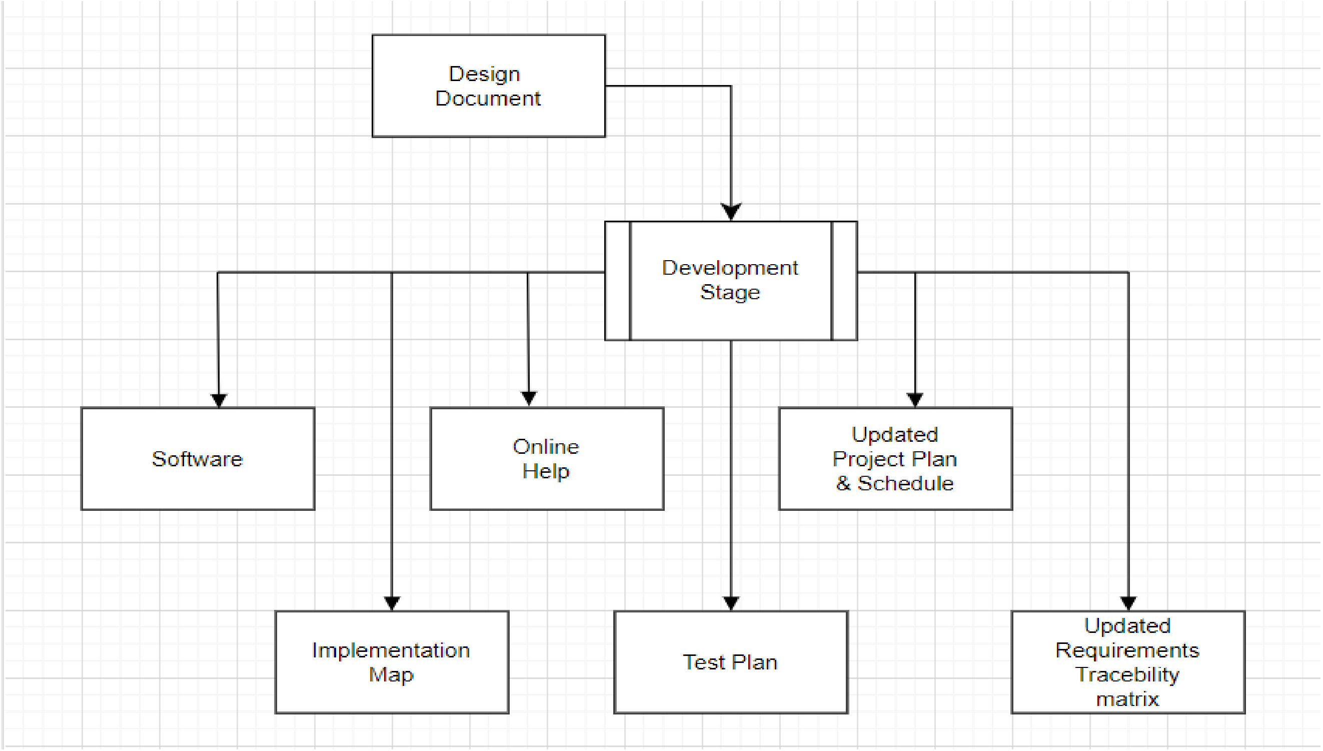
document, the RTM, and an updated project plan.

Design Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

Development Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



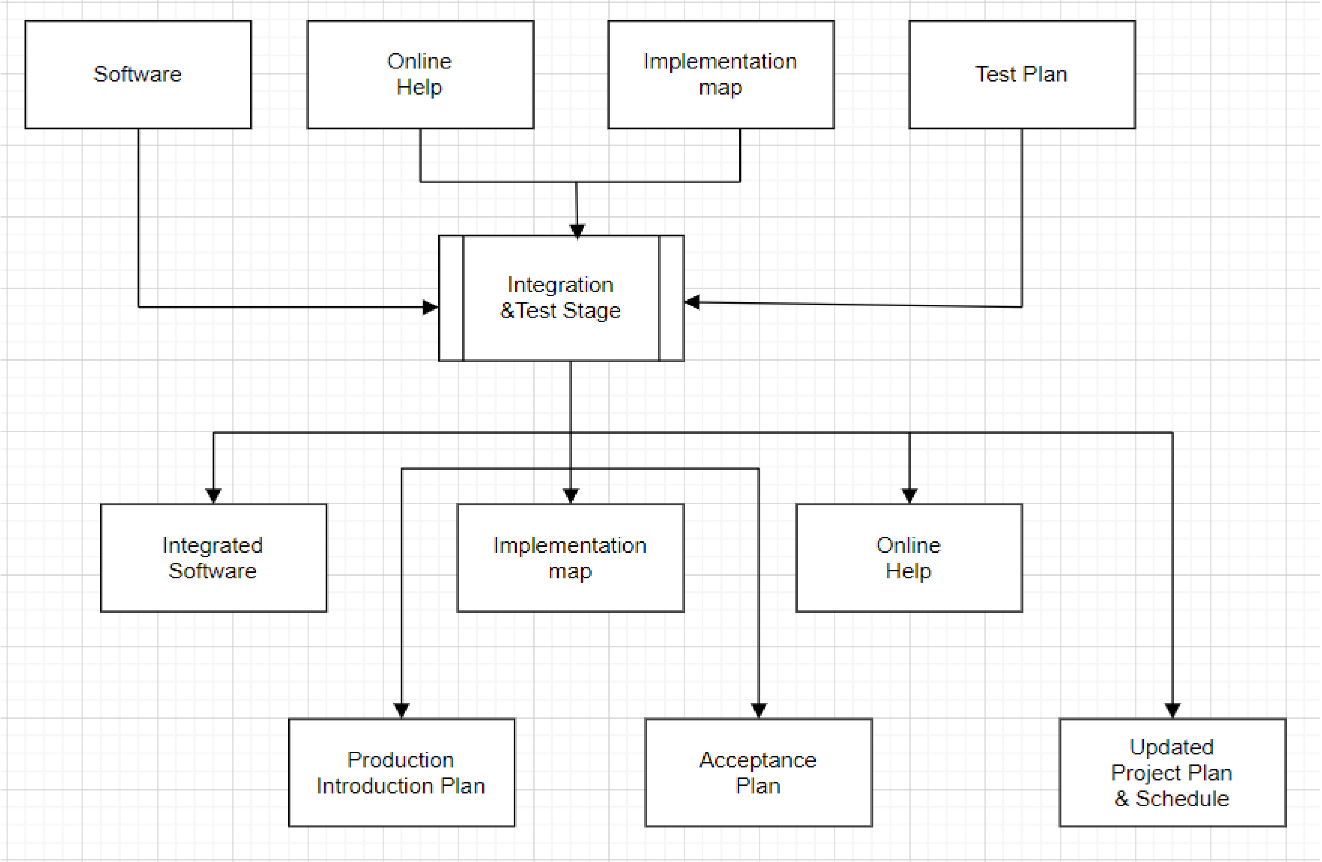
The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.

During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into

the Production Initiation Plan.

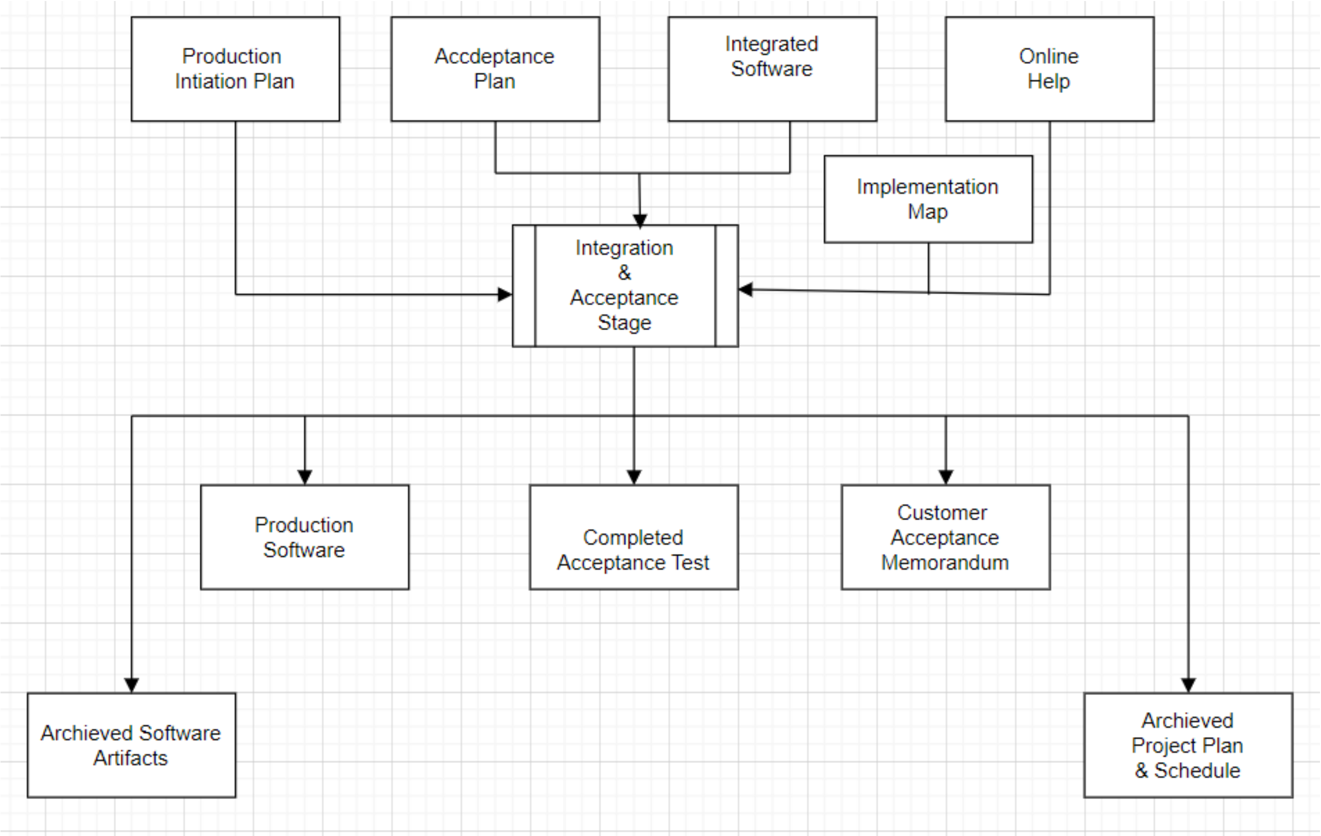


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Stage

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



Production Intiation Plan
Accdeptance Plan
Integrated Software
Online Help
Implementation Map
Integration & Acceptance Stage
Production Software
Completed Acceptance Test
Customer Acceptance Memorandum
Archieved Software Artifacts
Archieved Project Plan & Schedule

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving

3.3 STUDY OF THE SYSTEM

The data owner or a trusted third party generates the re-encryption key. A proxy runs the re-encryption algorithm with the key and revamps the cipher text before sending the new cipher text to the user. An intrinsic trait of a PRE scheme is that the proxy is not fully trusted (it has no idea of the data owner's secret key). This is seen as a prime candidate for delegating access to encrypted data in a secured manner, which is a crucial component in any data-sharing scenario. In addition, PRE allows for encrypted data in the cloud to be shared to authorized users while maintaining its confidentiality from illegitimate parties. Data disclosures can be minimized through the use of encryption since only users delegated by the data owner can effectively access the outsourced data.

3.4 FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibility that have to be analyzed are

3.4.1 ECONOMICAL FEASIBILITY

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

3.4.2 OPERATIONAL FEASIBILITY

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

3.4.3 TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and Web Logic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

3.4.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence

3.5 EXISTING SYSTEM

Park [16] provided a modification to the scheme in [15], where collusion between the service provider and revoked users is avoided. Their scheme was to basically replace the service provider with a trusted third party, which implies that there should be reliance on stronger trust assumption. Other schemes [17]–[19] have made similar approaches but utilized cipher text-policy ABE (CP-ABE) rather, in which the access policy is associated with the cipher text instead of the secret keys. also proposed a time-constrained access control scheme based on PRE and ABE. ABE was used to design time-based access control policies while PRE was used to update the time attributes. Although these schemes have their advantages, they are not suitable in the context of IOT due to the heavy computations on encryption and decryption.

An IBE PRE scheme suitable for data sharing was presented by Han *et al.* in [21]. The re-encryption keys were not only bound to the users' identities but also to a specific cipher text. This implied that the data owner had to create a different re encryption key for each pair of data user and shared file. A similar idea was proposed by Lin *et al.* [22] where they used a hierarchical PRE instead of an identity-based PRE. These two schemes tend to be inefficient when multiple and complex data pieces are considered. An identity-based broadcast encryption (IBBE) combined with PRE was proposed by in [23] for data sharing. Their scheme was a hybrid one that allowed the conversion to be done between the two protocols without

leaking any sensitive information.Wang *et al.* [24] also designed an identity-based PRE (IBPRE) scheme for accessing health records. The scheme achieved coarse-grained access control.

If a proxy receives the re-encryption key from the data owner, either all the cipher texts can be re-encrypted and accessible to the intended users or none at all. On that note, Shao *et al.* [25] proposed an IBEPRE scheme that is based on conditions. In their proposal, the proxy could transform a subset of cipher texts under an identity to other cipher texts under another identity. However, decryption rights to a group of users could not be authorized. In addition to the above, PRE has been used to mitigate security problems in IoT [26].

Zyskind *et al.* [27] used block chain to provide distributed personal data management and ensure privacy as well. The block chain was utilized as an automatic access control manager, and, hence, no third party was required. Only the data address was stored on the block chain and a distributed hash table was used as the implementation of the data storage. This reduced the risk of data leakage.

Disadvantages

1) The system was not implemented Attribute Based Encryption Mentod which leads less security on outsourced data.

2) The system is less security due to lack of Identity-Based Encryption.

3.6 PROPOSED SYSTEM

- The system proposes a secure access control framework to realize data confidentiality, and fine-grained access to data are achieved. This will also guarantee data owners' complete control over their data.
- The system gives a detailed description of our PRE scheme and the actualization of a complete protocol that guarantees security and privacy of data.
- To improve data delivery and effectively utilize the network bandwidth, edge devices serve as proxy nodes and perform re-encryption on the cached data. The edge devices are assumed to have

enough computation capabilities than the IOT devices and as such provide high performance networking.

- The security analysis of our scheme is presented, and we also test and compare its performance with existing schemes.

Advantages

1) The proposed system is secure against man-in-the-middle (MITM) attacks. MITM attacks get to the certificate authority (CA) to provide the user with forged public keys.

2) The proposed system finds Data Tampering and blocks when hackers compromise a system, they inject their own versions of the data into the system.

3.7 SOFTWARE REQUIREMENT SPECIFICATION

3.7.1 INTRODUCTION

A Software Requirements Specification (SRS) – a requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non- functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System Requirements specification:

A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- **Business requirements** describe in business terms what must be delivered or accomplished to provide value.

- **Product requirements** describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- **Process requirements** describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement);requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development styles.

3.7.2 PURPOSE

An systems engineering, a **requirement** can be a description of *what* a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Non-functional requirements, or 'performance requirements' or 'quality of service requirements.' Examples of such requirements include usability, availability, reliability, sup portability, test ability and maintainability.

A collection of requirements defines the characteristics or features of the desired system. A 'good' list of requirements as far as possible avoids saying *how* the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called "implementation bias" or "solution engineering". However, implementation constraints on the solution may validly be expressed by the future owner, for example for required interfaces to external systems; for interoperability with other systems; and for commonality (e.g. of user interfaces) with other owned products.

In software engineering, the same meanings of requirements apply, except that the focus of interest is the software itself.

3.7.3 FUNCTIONAL REQUIREMENTS

Functional requirements define the basic system behaviour. Essentially, they are **what** the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviour and include calculations, data input, and business processes.

Input
System Behaviour
Output

Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product **features** and focus on user **requirements**.

3.7.4 MODULES INVOLVED

- **Data Owner Module**

  In this module, the data owner uploads their data in the public cloud server. For the security purpose the data owner encrypts the data file and assigns the digital sign and then store in the cloud. The data owner can check the data integrity of the file over Corresponding cloud server. The Data owner can have capable of manipulating the encrypted data file and data owner can update the file contents as well as delete his own file.

- **Key Generation Centre**

  In this module, the KGC Generates the Secret Key requested by the data user, the KGC checks the file if present generates the appropriate Secret Key. The KG-CSP allows viewing the Secret Key generated files and also the transactions related to the file.

- **Proxy Server**

  The server will manage and authorize Users and maintain all data transactions between data owner and cloud server, end user.

- **Data User Module**

  In this module, Data user logs in by using his user name and password. After he will request for secret key of required file from **CSP**, and get the secrete key from KGC. After getting secrete key he is trying to download file by entering file name and secrete key from cloud server.

- **Data Encryption and Decryption**

  All the legal users in the system can freely query any interested encrypted and decrypt ed data. Upon receiving the data from the server, the user runs the decryption algorithm Decrypt to decrypt the cipher text by using its secret keys from different Users. Only the attributes the user possesses satisfy the access structure defined in the cipher text CT, the user can get the content.

- **Attacker Module**

  In Data user module, while downloading time if remote user enters wrong trapdoor or secrete key then he is treated as Digital sign attacker or Secret Key attacker.

- **Data Integrity Check**

  Data will be verified in the cloud to check it is integrated by attacker or not. If it is integrated then it is recovering from the data owner.

3.7.5 NON-FUNCTIONAL REQUIREMENTS

- Flexibility & Scalability Oracle itself has given a set of applications with JDK but the whole developer community can develop their own applications and they have access to same resources and public API which are accessible to core applications.
- Robust The application is fault tolerant with respect to illegal user/receiver inputs. Error checking has been built in the system to prevent system failure.
- Fragmentation Java gave the same environment which is open; the entire A Pi's which is open to all the devices which reduces fragmentation. If you develop an java application, it will run on all the devices.
- **Open Source**:Java open source is free and easy to download. Java is a platform in depended based programming language and The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine.
- **Scalability**: The system can be extended to integrate the modifications done in the present application to improve the quality of the product. This is meant for the future works that is to be done on the application.

- **Reliability**:Since the application is being developed through java, the most famous, efficient and reliable language, so it is reliable in every aspect until and unless there is an error in the programming side. Thus the application can be a compatible and reliable one.
- **Portability**:This System must be intuitive enough such that user with average background in using mobile phones can quickly experiment with the system and learn how to use the project. The system has user friendly interface.

3.7.6 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

- **Processor :** Intel i3 11th Generation
- **Hard Disk : 1**28 GB.
- **Ram :** 4 GB.

SOFTWARE REQUIREMENTS:

- **Operating system :** Windows 10
- **Coding Language :** Java,J2EE(JSP,Servlet,Java Bean)
- **Front-End :** HTML5,CSS,JS.
- **Data Base :** MySQL.

4. SYSTEM DESIGN

4.1 INTRODUCTION

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

4.2 NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, upgradation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletion anomaly: Unintended loss of data due to deletion of other data.

Update anomaly: Data inconsistency resulting from data redundancy and partial update **Normal Forms**: These are the rules for structuring relations that eliminate anomalies. **FIRST NORMAL FORM**:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1) Primary key is a not a composite primary key
2) No non key attributes are present
3) Every non key attribute is fully functionally dependent on full set of primary key.

THIRD NORMAL FORM:

A relation is said to be in third normal form if their exits no transitive dependencies.

Transitive Dependency: If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

4.3 ER DIAGRAM

- The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the personalities that are necessary for the system state to continue.
- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object

descriptions.

- The set of primary components that are identified by the

1. Data object □Relationships
2. Attributes □ Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.

DATA FLOW DIAGRAMS:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Your don, Gena and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and data flow names have the first letter of each work capitalized

ADFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out. Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

4.4 UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- Actors
- Business processes
- (logical)components
- Activities
- programming language statements
- database schema,and
- Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies· UML has synthesized the notations of the Bloch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

4.4 USE CASE DIAGRAM

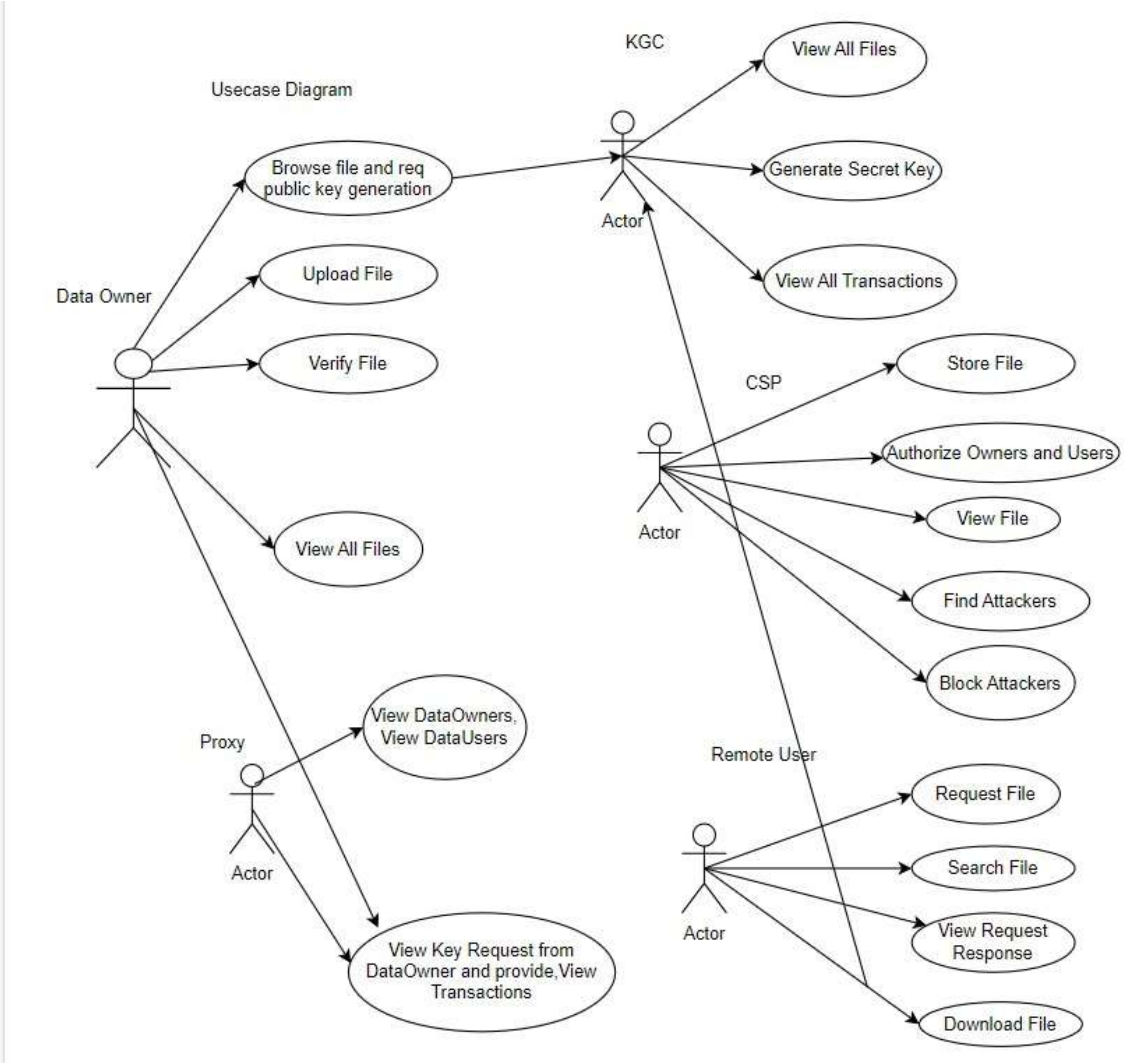
A Use Case Model describes the proposed functionality of a new system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system. This interaction is a single unit of meaningful work, such as Create Account or View Account Details.

Each Use Case describes the functionality to be built in the proposed system, which can include another Use Case's functionality or extend another Use Case with its own behavior.

4.4.1 Use-Case diagram

A use case illustrates a unit of functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases.

Use-case diagrams generally show groups of use cases — either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality (e.g., all security administration-related use cases).



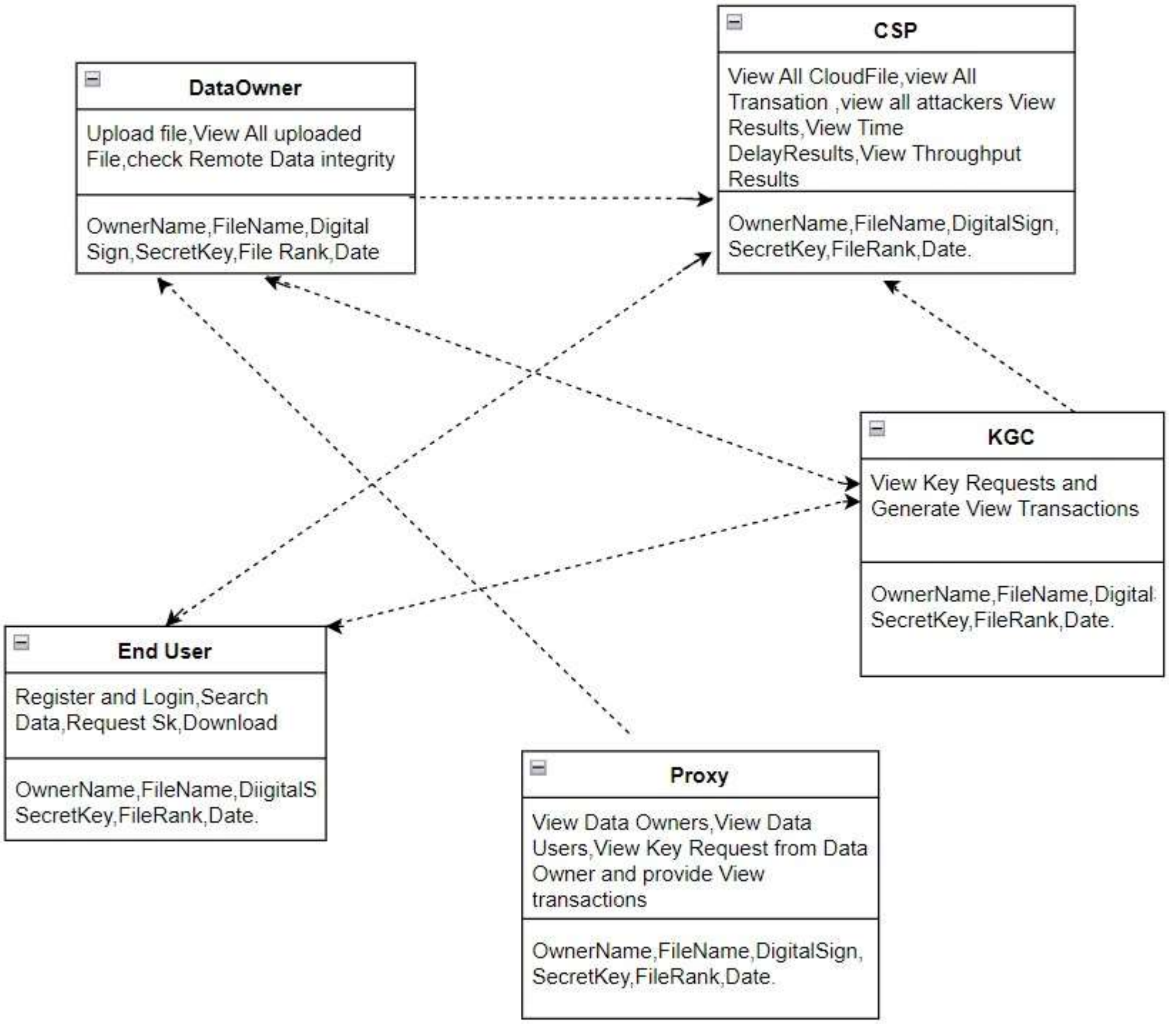
4.4.2 CLASS DIAGRAM

The class diagram shows how the different entities (people, things, and data) relate to each other; in other words, it shows the static structures of the system. A class diagram is diagrams that show a set of classes, interfaces and collaborations and then relationship. Graphically a class diagram is a collection of vertices and arcs.

A class diagram commonly contains the following

- Class
- Interfaces

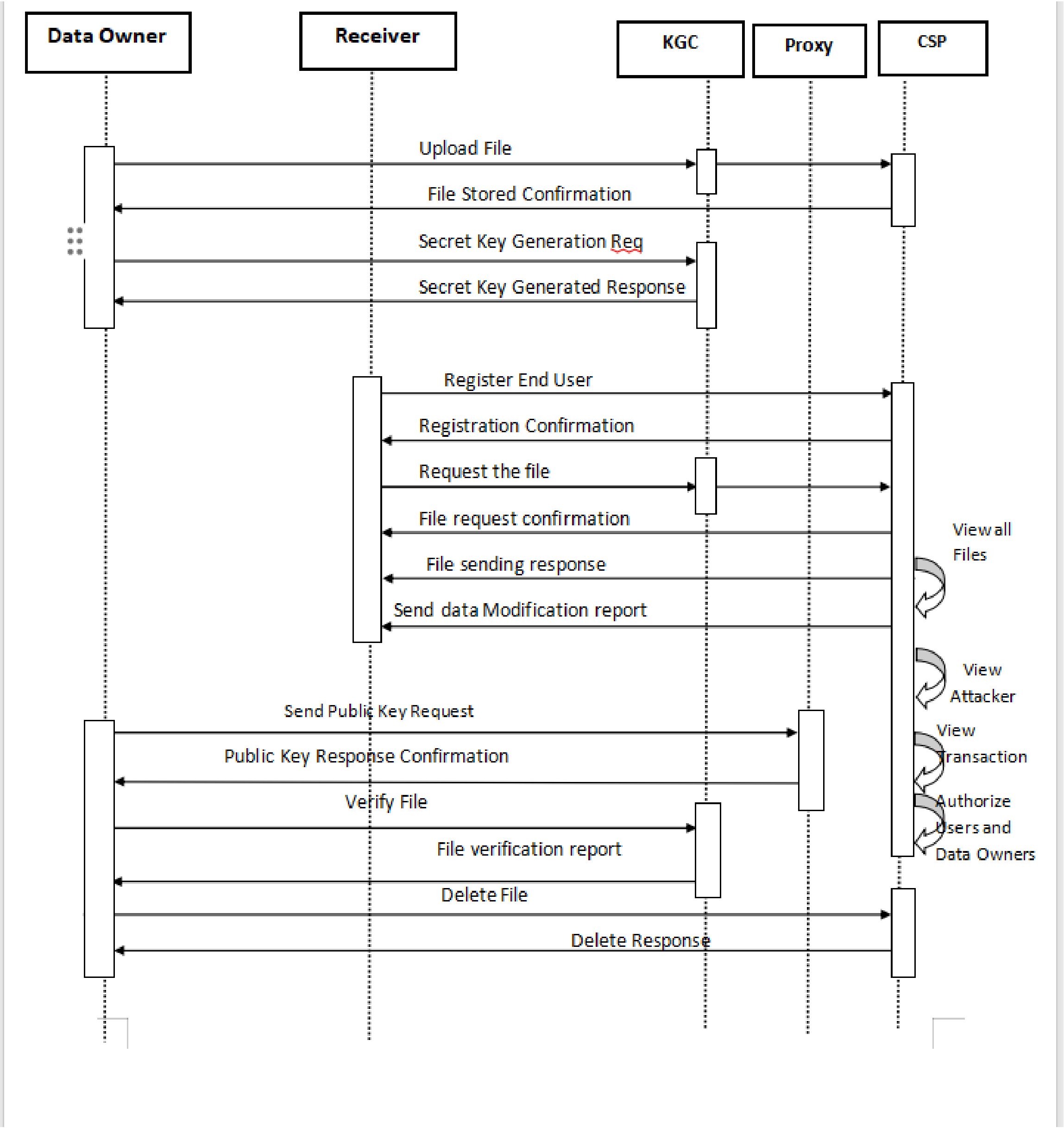
- Dependency
- Generalization
- Association Relationship


CSP
View All CloudFile,view All Transation ,view all attackers View Results,View Time DelayResults,View Throughput Results
OwnerName,FileName,DigitalSign, SecretKey,FileRank,Date.
DataOwner
Upload file,View All uploaded File,check Remote Data integrity
OwnerName,FileName,Digital Sign,SecretKey,File Rank,Date
KGC
View Key Requests and Generate View Transactions
OwnerName,FileName,Digital SecretKey,FileRank,Date.
End User
Register and Login,Search Data,Request Sk,Download
OwnerName,FileName,DiigitalS SecretKey,FileRank,Date.
Proxy
View Data Owners,View Data Users,View Key Request from Data Owner and provide View transactions
OwnerName,FileName,DigitalSign, SecretKey,FileRank,Date.

4.4.3 SEQUENCE DIAGRAM

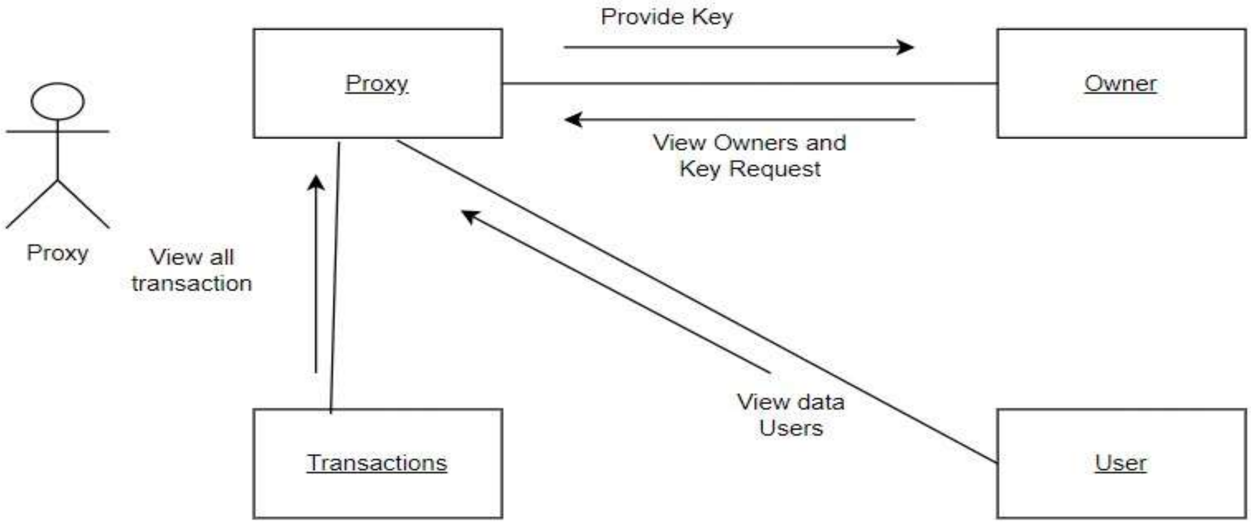
Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects. A sequence diagram has two dimensions:

The vertical dimension shows the sequence of messages/calls in the time order that they occur; the horizontal dimension shows the object instances to which the messages are sent.



4.4.4 COLLABORATION DIAGRAM

Collaboration diagrams are Interaction diagrams. They convey the same information as sequence diagram, but they focus on object roles instead of the times that messages are sent

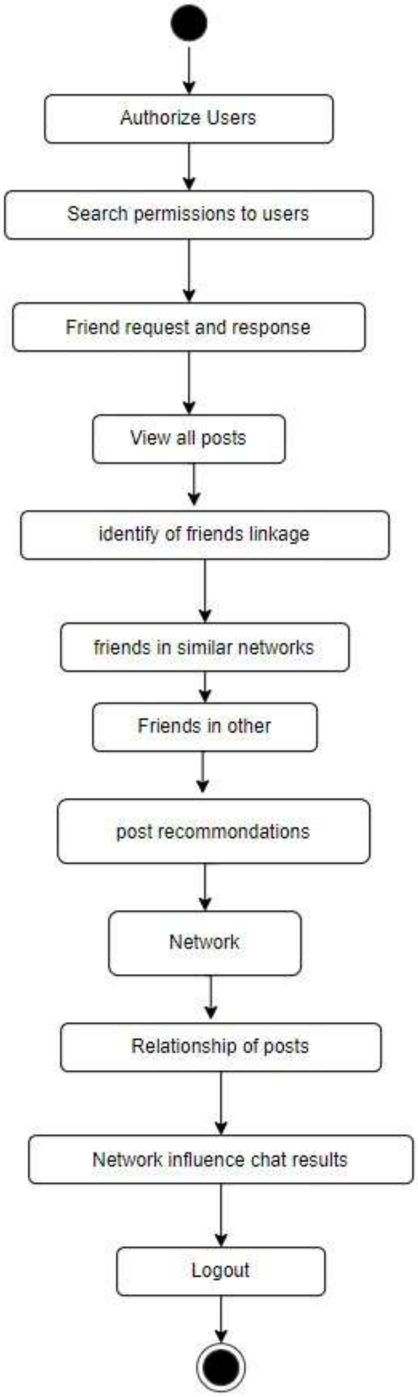


4.4.5 ACTIVITY DIAGRAM

Activity diagrams show the procedural flow of control between two or more class objects while processing an activity. Activity diagrams can be used to model higher-level business process at the business unit level, or to model low-level internal class actions.

In my experience, activity diagrams are best used to model higher-level processes, such as how the company is currently doing business, or how it would like to do business.

This is because activity diagrams are "less technical" in appearance, compared to sequence diagrams, and business-minded people tend to understand them more quick


Authorize Users
Search permissions to users
Friend request and response
View all posts
identify of friends linkage
friends in similar networks
Friends in other
post recommondations
Network
Relationship of posts
Network influence chat results
Logout

4.4.6 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Thus, from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

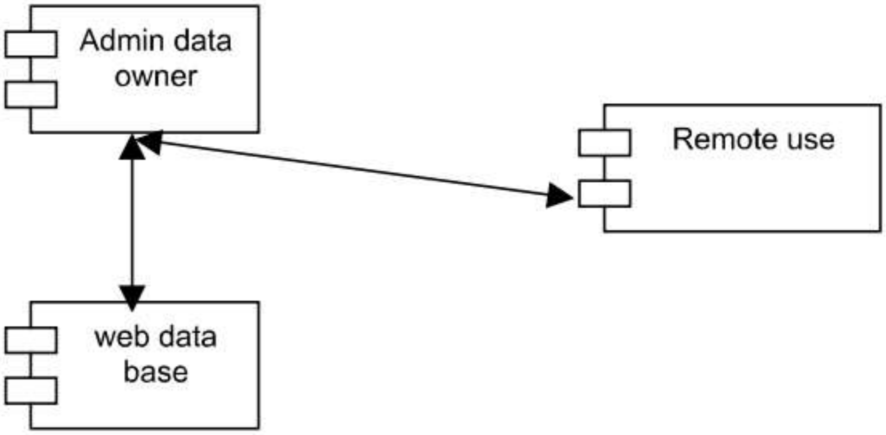
Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

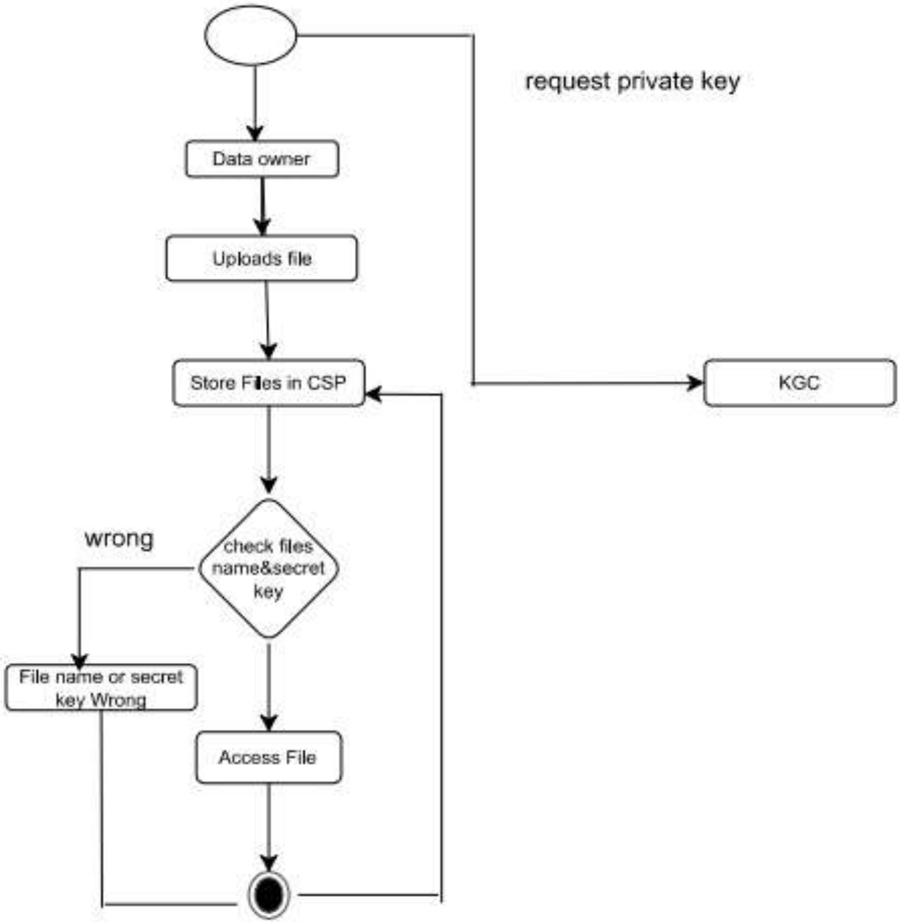
The purpose of the component diagram can be summarized as

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

•

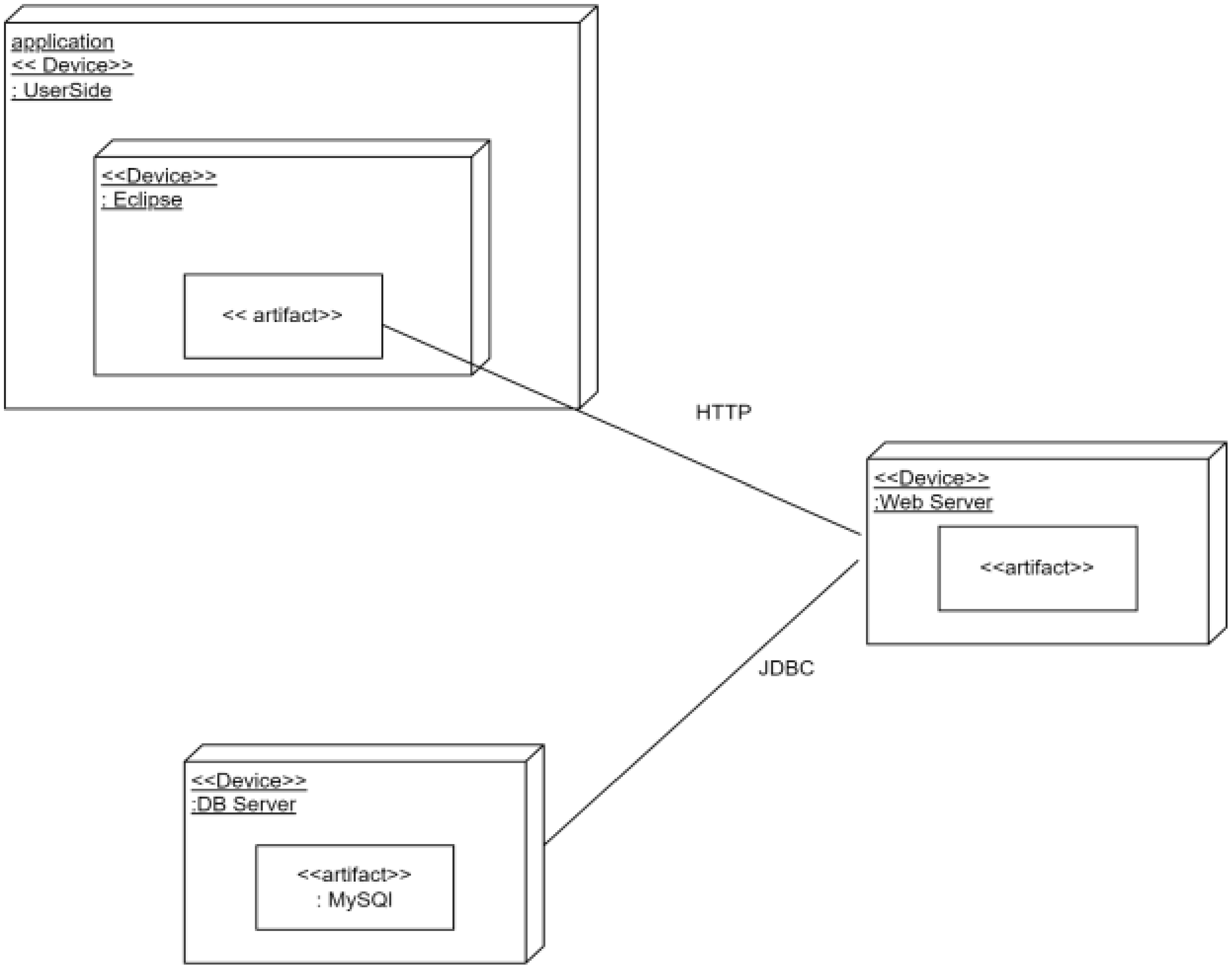


4.4.7 STATE DIAGRAM



-

4.4.8 DEPLOYMENT DIAGRAM



5. IMPLEMENTATION

5.1 INTRODUCTION

Implementation is the process of converting a new or revised system design into operational one. There are three types of Implementation:

Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, and verifying printouts for integrity.

Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems.

Implementation of a modified application to replace an existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

Implementation in Generic tool project is done in all modules. In the first module User level identification is done. In this module every user is identified whether they are genuine one or not to access the database and also generates the session for the user. Illegal use of any form is strictly avoided.

In the Table creation module, the tables are created with user specified fields and user can create many table at a time. They may specify conditions, constraints and calculations in creation of tables. The Generic code maintain the user requirements throughout the project.

In Updating module user can update or delete or Insert the new record into the database. This is very important module in Generic code project. User has to specify the filed value in the form then the Generic tool automatically gives whole filed values for that particular record.

In Reporting module user can get the reports from the database in 2Dimentional or 3Dimensional view. User has to select the table and specify the condition then the report will be generated for the user.

5.2 ALGORITHMS

5.2.1 SHA-256(SECURED HASH ALGORITHM)

The Secure Hash Algorithm (SHA) is a family of cryptography hash functions that are widely used in various security applications and protocols. SHA algorithms generate a fixed-size hash value (typically 160 or 256 bits) from input data of any size, making them suitable for tasks like digital

5.2.2 ADVANCED ENCRYPTION STANDARD (AES)

The Advanced Encryption Standard (AES) is a symmetric block cipher algorithm that is widely used for encrypting and decryption data. It was selected by the U.S. National Institute of Standards and Technology (NIST) in 2001 as the standard encryption algorithm for securing sensitive information.AES operates on fixed-size blocks of data, with a block size of 128 bits. It supports key sizes of 128, 192, or 256 bits, with the larger key sizes providing stronger security. The algorithm consists of several rounds of substitution, permutation, and mixing operations, designed to ensure the confidentiality and integrity of the encrypted data.

5.2.3 RSA Algorithm

RSA algorithm is a widely used asymmetric cryptography algorithm that is used for encryption and digital signatures. While it's not typically used to generate secret keys directly, it can be used as part of a key exchange process to establish a shared secret key between two parties.In RSA, each party 1)Key Generation: Each party generates a public-private key pair.

5.2.4 ID-PUIC PROTOCOL (IDENTITY-BASED PROXY-ORIENTED DATA UPLOADING AND REMOTE DATA INTEGRITY CHECKING)

1) Key Generation: The cloud service provider (CSP) generates a master public-private key pair for the identity-based encryption (IBE) scheme. The private key is kept secret, and the public key is used for encryption.

5.3 SAMPLE CODE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-aller.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style1 {font-size: 14px}

.style2 {font-size: 24px}

.style5 {font-size: 14px; color: #FFFF00; }

.style6 {color: #FFFF00}

-->

</style>

</head>

<body>

<div class="main">

  <div class="header">

    <div class="header_resize">

      <div class="menu_nav">

        <ul>
        <li class="active"><a href="index.html"><span>Home Page</span></a></li>
                <li><a href="client.html">Data Owner </a></li>
                <strong></strong>
        <li><a href="pcs.html">CSP</a></li>
        <li><a href="proxy.html"><span>Proxy</span></a></li>
        <li><a href="kgc.html"><span>TA</span></a></li>
        <li><a href="enduser.html"><span>Data User </span></a></li>
        </ul>
      </div>
      <div class="logo">
        <h1 class="style1"><a href="index.html" class="style2">A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain</a></h1>
      </div>


      </div>
      <div class="clr"></div>
    </div>
  </div>
  <div class="content">
```

6. SYSTEM TESTING

6.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Software Testing is the process of confirming the functionality and correctness of software by running it. Software testing is usually performed for one of two reasons:

1) Defect detection
2) Reliability estimation.

White box testing is concerned only with testing the software product; it cannot guarantee that the complete specification has been implemented. Black box testing is concerned only with testing the specification; it cannot guarantee that all parts of the implementation have been tested. Thus black box testing is testing against the specification and will discover faults of omission, indicating that part of the specification has not been fulfilled. White box testing is testing against the implementation and will discover faults of commission, indicating that part of the implementation is faulty. In order to fully test a software product both black and white box testing are required

The problem of applying software testing to defect detection is that software can only suggest the presence of flaws, not their absence (unless the testing is exhaustive). The problem of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed. In both of these cases, the mechanism used to determine whether program output is correct is often impossible to develop. Obviously the benefit of the entire software testing process is highly dependent on many different pieces. If any of these parts is faulty, the entire process is compromised.

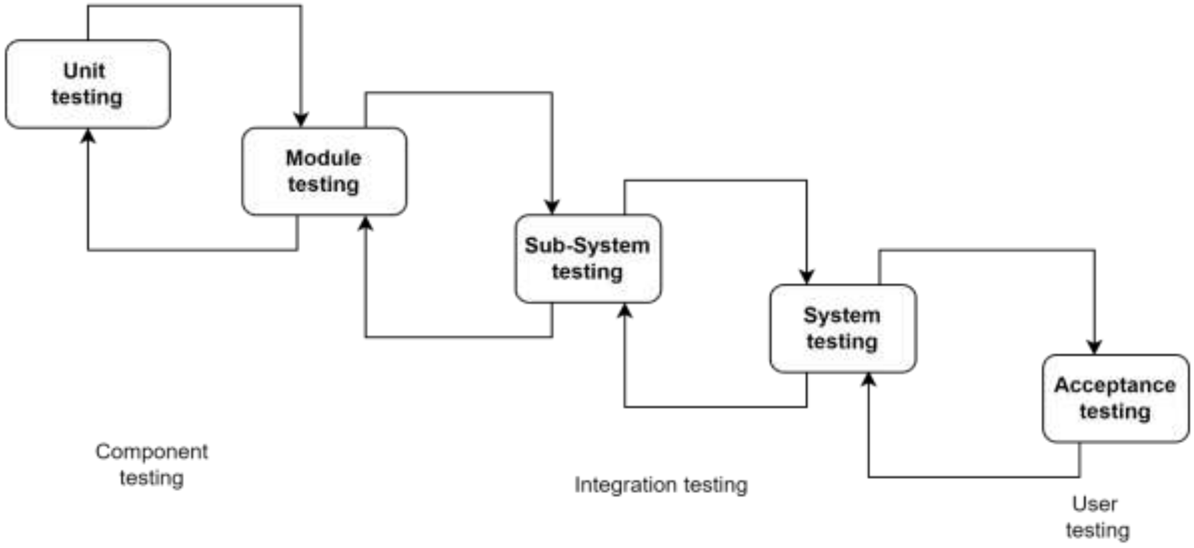
Software is now unique unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical

The whole computer based system is checked not only for validity but also to meet the objectives.

6.2 STRATEGIC APPROACH TO SOFTWARETESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirements analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decreases the level of abstraction on each item.

A Strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing will progress by moving outward along the spiral to integration testing, where the focus on the design and the concentration of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed . Finally we arrive at system testing, where the software and other system elements are tested as a whole.



Different Levels of Testing:

Client Needs	Acceptance Testing
Requirements	System Testing
Design	Integration Testing
Code	Unit Testing

Testing is the process of finding difference between the expected behavior specified by system models and the observed behavior of the implemented system.

Testing Activities:

Different levels of testing are used in the testing process, each level of testing aims to test different aspects of the system. the basic level sare:

1. **Unit Testing:**

Unit testing focuses on the building blocks of the software system, that is, objects and sub system. There are three motivations behind focusing on components. First, unit testing reduces the complexity of the overall tests activities, allowing us to focus on smaller units of the system. Second, unit testing makes it easier to pinpoint and correct faults given that few components are involved in this test. Third, Unit testing allows parallelism in the testing activities, that is each component can be tested independently of one another. Hence the goal is to test the internal logic of the module.

2. Integration Testing:

In the integration testing, many test modules are combined into sub systems, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis being on testing module interaction.

After structural testing and functional testing we get error free modules. These modules are to be integrated to get the required results of the system. After checking a module, another module is tested and is integrated with the previous module. After the integration, the test cases are generated, and the results are tested.

3. System Testing:

In system testing the entire software is tested. The reference document for this process is the requirement document and the goal is to see whether the software meets its requirements. The system was tested for various test cases with various inputs.

4. Acceptance Testing:

Acceptance testing is sometimes performed with realistic data of the client to demonstrate that the software is working satisfactory. Testing here focus on the external behavior of the system, the internal logic of the program is not emphasized. In acceptance testing the system is tested for various inputs.

Types of Testing:

1. Black box or functional testing
2. White box testing or structural testing

1.Black box testing:

This method is used when knowledge of the specified function that a product has been designed to perform is known. The concept of black box is used to represent a system whose inside workings are not available to inspection. In a black box the test item is a "Black", since its logic is unknown, all that is known is what goes in and what comes out, or the input and output.

Black box testing attempts to find errors in the following categories: Incorrect or missing functions

Interface errors

Errors in data structure

Performance errors

Initialization and termination errors

As shown in the following figure of Black box testing , we are not thinking of the internal workings , just we think about

What is the output to our system?

What is the output for given input to our system?

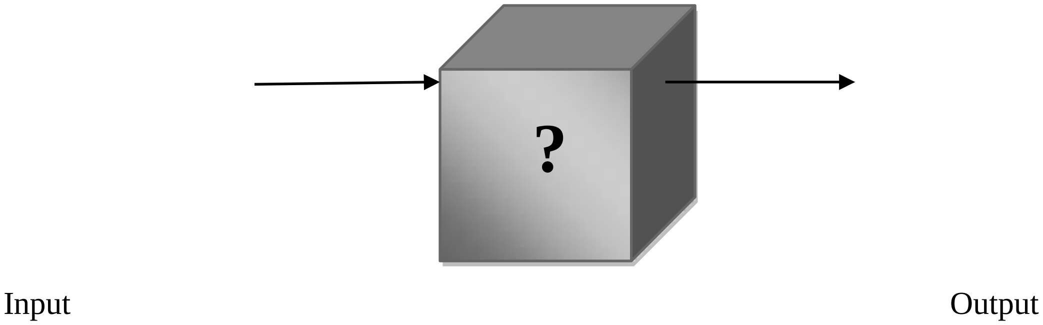


Fig:6.2.2 The Black box is an imaginary box that hides its internal workings

2.White box testing:

White box testing is concerned with testing the implementation of the program. the intent of structural is not to exercise all the inputs or outputs but to exercise the different programming and data structure used in the program. Thus structural testing aims to achieve test cases that will force the desire coverage of different structures. Two types of path testing are statement testing coverage and branch testing coverage.

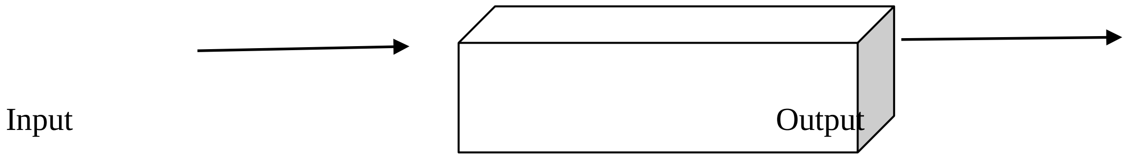


Fig:6.2.3 White Box testing strategy, the internal workings

Test Plan:

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources , and specified guidelines for testing . During the testing of the unit the specified test cases are executed and the actual result compared with expected output. The final output of the testing phase is the test report and the error report.

Test Data:

Here all test cases that are used for the system testing are specified. The goal is to test the different functional requirements specified in Software Requirements Specifications (SRS) document.

Unit Testing:

Each individual module has been tested against the requirement with some test data.

Test Report:

The module is working properly provided the user has to enter information. All data entry forms have tested with specified test cases and all data entry forms are working properly.

7. TECHNOLOGY DESCRIPTION AND CODING

7.1 INTRODUCTION TO JAVA

Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multi threaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

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

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be abit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

- **Software components**: Known as JavaBeans^{TM}, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC^{TM})**: Provides uniform access to a wide range of relational databases.

Java Server Pages

Java Server Pages (JSP) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. JSP have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

What is JavaServer Pages?

JavaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through Web page forms, present records from a database or another source, and create Web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing Java Beans components, passing control between pages, and sharing information between requests, pages etc.

Why Use JSP?

Java Server Pages often serve the same purpose as programs implemented using the **Common Gateway Interface (CGI)**. But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- Java Server Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including **JDBC, JNDI, EJB, JAXP,** etc.
- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

Advantages of JSP

Following table lists out the other advantages of using JSP over other technologies −

vs. Active Server Pages (ASP)

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

vs. Pure Servlets

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

vs. Server-Side Includes (SSI)

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

vs. JavaScript

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

vs. Static HTML

Regular HTML, of course, cannot contain dynamic information.

JSP - Environment Setup

A development environment is where you would develop your JSP programs, test them and finally run them.

This tutorial will guide you to setup your JSP development environment which involves the following steps −

Setting up Java Development Kit

This step involves downloading an implementation of the Java Software Development Kit (SDK) and setting up the PATH environment variable appropriately.

You can download SDK from Oracle's Java site − Java SE Downloads.

Once you download your Java implementation, follow the given instructions to install and configure the setup. Finally set the **PATH and JAVA_HOME** environment variables to refer to the directory that contains **java** and **javac**, typically **java_install_dir/bin** and **java_install_dir** respectively.

If you are running Windows and install the SDK in **C:\jdk10.0.2** , you need to add the following line in your **C:\autoexec.bat** file.

```
set PATH = C:\jdk1.5.0_20\bin;%PATH%
set JAVA_HOME = C:\jdk1.5.0_20
```

Alternatively, on **Windows XP/10**, you can also right-click on **My Computer**, select **Properties**, then **Advanced**, followed by **Environment Variables**. Then, you would update the PATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if the SDK is installed in **/usr/local/jdk10.0.2** and you use the C shell, you will put the following into your **.cshrc** file.

```
setenv PATH /usr/local/jdk10.0.2/bin:$PATH
setenv JAVA_HOME /usr/local/jdk10.0.2
```

Alternatively, if you use an **Integrated Development Environment (IDE)** like Borderland **J Builder, Eclipse, IntelliJ IDEA**, or **Sun ONE Studio**, compile and run a simple program to confirm that the IDE knows where you installed Java.

Setting up Web Server: Tomcat

A number of Web Servers that support Java Server Pages and Servlets development are available in the market. Some web servers can be downloaded for free and Tomcat is one of them.

Apache Tomcat is an open source software implementation of the JavaServer Pages and Servlet technologies and can act as a standalone server for testing JSP and Servlets, and can be integrated with the Apache Web Server. Here are the steps to set up Tomcat on your machine −

- Download the latest version of Tomcat from https://tomcat.apache.org/.
- Once you downloaded the installation, unpack the binary distribution into a convenient location. For example, in **C:\apache-tomcat-9.0 on windows, or /usr/local/apache-tomcat-9.0** on Linux/Unix and create **CATALINA_HOME** environment variable pointing to these locations.

Tomcat can be started by executing the following commands on the Windows machine −

```
%CATALINA_HOME%\bin\startup.bat

or

C:\apache-tomcat-9.0\bin\startup.bat
```

Tomcat can be started by executing the following commands on the Unix (Solaris, Linux, etc.) machine −

```
$CATALINA_HOME/bin/startup.sh

or

/usr/local/apache-tomcat-9.0/bin/startup.sh
```

After a successful startup, the default web-applications included with Tomcat will be available by visiting **http://localhost:8080/**.

Upon execution, you will receive the following output −

Further information about configuring and running Tomcat can be found in the documentation included here, as well as on the Tomcat web site − https://tomcat.apache.org/.

Tomcat can be stopped by executing the following commands on the Windows machine −

```
%CATALINA_HOME%\bin\shutdown
or
C:\apache-tomcat-5.5.29\bin\shutdown
```

Tomcat can be stopped by executing the following commands on Unix (Solaris, Linux, etc.) machine −

```
$CATALINA_HOME/bin/shutdown.sh

or

/usr/local/apache-tomcat-5.5.29/bin/shutdown.sh
```

Setting up CLASSPATH

Since servlets are not part of the Java Platform, Standard Edition, you must identify the servlet classes to the compiler.

If you are running Windows, you need to put the following lines in your **C:\autoexec.bat** file.

```
set CATALINA = C:\apache-tomcat-5.5.29
set CLASSPATH = %CATALINA%\common\lib\jsp-api.jar;%CLASSPATH%
```

Alternatively, on **Windows NT/2000/XP**, you can also right-click on **My Computer**, select **Properties**, then **Advanced**, then **Environment Variables**. Then, you would update the CLASSPATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if you are using the C shell, you would put the following lines into your **.cshrc** file.

```
setenv CATALINA = /usr/local/apache-tomcat-5.5.29
setenv CLASSPATH $CATALINA/common/lib/jsp-api.jar:$CLASSPATH
```

NOTE − Assuming that your development directory is **C:\JSPDev (Windows)** or **/usr/JSPDev (Unix)**, then you would need to add these directories as well in CLASSPATH.

The Life cycle of a JSP Page

The JSP pages follow these phases:

- Translation of JSP Page

- Compilation of JSP Page
- Class loading (the class loader loads class file)
- Instantiate (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jsp Service() method).
- Destroy (the container invokes jsp Destroy() method).

As depicted in the above diagram, JSP page is translated into Servlet by the help of JSP translator. The JSP translator is a part of the web server which is responsible for translating the JSP page into Servlet. After that, Servlet page is compiled by the compiler and gets converted into the class file. Moreover, all the processes that happen in Servlet are performed on JSP later like initialization, committing response to the browser and destroy.

Do I need to follow the directory structure to run a simple JSP?

No, there is no need of directory structure if you don't have class files or TLD files. For example, put JSP files in a folder directly and deploy that folder. It will be running fine. However, if you are using Bean class, Servlet or TLD file, the directory structure is required.

The Directory structure of JSP

The directory structure of JSP page is same as Servlet. We contain the JSP page outside the WEB-INF folder or in any directory.

Creating JSP in Eclipse IDE with Tomcat server

- Create a Dynamic web project
- create a jsp
- start tomcat server and deploy the project

1) Create the dynamic web project

For creating a dynamic web project click on File Menu -> New -> dynamic web project -> write your project name e.g. first -> Finish.

2) Create the JSP file in eclipse IDE

For creating a jsp file explore the project by clicking the + icon -> right click on WebContent -> New -> jsp -> write your jsp file name e.g. index -> next -> Finish.

Now JSP file is created, let's write some code.

3) Start the server and deploy the project:

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

you are using Eclipse IDE first time, you need to configure the tomcat server First. Click for How to configure tomcat server in eclipse IDE

Now start the tomcat server and deploy project

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> add All -> finish.

Yes, Let's see JSP is successfully running now.

What is MySQL?

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, salable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is ***My Ess Que Ell***. *However, you can pronounce it in your way*. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, Mac OS, etc. with C, C++, and Java languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to updates the table indexes automatically.

- It uses many SQL queries and combines useful information from multiple tables for the end-users.

How MySQL Works?

MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model.

The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands. The working of MySQL database with MySQL Server are as follows:

1. MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.
2. Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
3. Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

A client can use any MySQL GUI. But, it is making sure that your GUI should be lighter and user-friendly to make your data management activities faster and easier. Some of the most widely used MySQL GUIs are MySQL Workbench, Sequel Pro, DB Visualizer, and the Navigate DB Admin Tool. Some GUIs are commercial, while some are free with limited functionality, and some are only compatible with MacOS. Thus, you can choose the GUI according to your needs.

Reasons for popularity

MySQL is becoming so popular because of these following reasons:

- MySQL is an open-source database, so you don't have to pay a single penny to use it.
- MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages.

- MySQL is customization because it is an open-source database, and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.
- MySQL is quicker than other databases, so it can work well even with the large data set.
- MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL is very friendly with PHP, the most popular language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

What is XAMPP?

XAMPP is an abbreviation where ***X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl***, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a back end scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/ programming languages such as **PHP** and **Perl**. X denotes Cross-platform, which means that it can work on different platforms such as Windows, Linux, and mac OS.

Many other components are also part of this collection of software and are explained below.

1. **Cross-Platform:** Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.
2. **Apache:** It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.
3. **MariaDB:** Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.
4. **PHP:** It is the back end scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for **Hypertext Processor**. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.
5. **Perl:** It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.
6. **Php My Admin:** It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.
7. **Open SSL:** It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol. Presently version 0.9.8 is a part of XAMPP.
8. **XAMPP Control Panel:** It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.2.1 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial.

9. **Webalizer:** It is a Web Analytic software solution used for User logs and provide details about the usage.
10. **Mercury:** It is a mail transport system, and its latest version is 4.62. It is a mail server, which helps to manage the mails across the web.
11. **Tomcat:** Version 7.0.42 is currently being used in XAMPP. It is a servlet based on JAVA to provide JAVA functionalities.
12. **File zilla:** It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files. Its recently updated version is 0.9.41.

Installing XAMPP

Our XAMPP tutorial will take you through the installation process for the software package on Windows. If you're using Linux or Mac OS X, then the steps listed below for the installation process may differ.

Step 1: Download

XAMPP is a release made available by the non-profit project Apache Friends. Versions with PHP 5.5, 5.6, or 7 are available for download on the Apache Friends website.

Step 2: Run .exe file

Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending .exe.

Step 3: Deactivate any antivirus software

Since an active antivirus program can negatively affect the installation process, it's recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed.

Before installing XAMPP, it is advisable to disable the anti-virus program temporarily

Step 4: Deactivate UAC

User Account Control (UAC) can interfere with the XAMPP installation because it limits writing access to the C: drive, so we recommend you deactivate this too for the duration of the installation process. To find out how to turn off your UAC, head to the Microsoft Windows support pages.

User account control can affect the installation of XAMPP

Step 5: Start the setup wizard

After you’ve opened the .exe file (after deactivating your antivirus program(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on ‘Next’ to configure the installation settings.

You can start the setup on the startup screen

Step 6: Choose software components

Under ‘Select Components’, you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click ‘Next’.

In the dialog window entitled 'select components', you can choose the software components before installation

Step 7: Choose the installation directory

In this next step, you have the chance to choose where you’d like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C:\ for you. After you’ve chosen a location, click ‘Next’.

For the next step, you need to select the directory where XAMPP should be installed

Step 8: Start the installation process

Once all the aforementioned preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process can take several minutes in total. You can follow the progress of this installation by keeping an eye on the green loading bar in the middle of the screen.

According to the default settings, the selected software components are unpacked and installed in the target folder

Step 9: Windows Firewall blocking

Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn’t recommended.

Step 10: Complete installation

Once all the components are unpacked and installed, you can close the setup wizard by clicking on ‘Finish’. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.

By clicking on 'finish', the XAMPP Setup Wizard is completed

The XAMPP Control Panel

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. **The clear user interface** logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

- **Config:** allows you to configure the XAMPP as well as the individual components
- Netstat**:** shows all running processes on the local computer
- **Shell:** opens a UNIX shell
- **Explorer:** opens the XAMPP folder in Windows Explorer
- **Services:** shows all services currently running in the background
- **Help:** offers links to user forums
- **Quit:** closes the XAMPP Control Panel

In the Control Panel, you can start and stop individual modules

Starting modules

Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under ‘Actions’. You can see which modules have been started because their names are highlighted green under the ‘Module’ title.

An active module is marked in green in the Control Panel

If a module can’t be started as a result of an error, you’ll be informed of this straight away in red font. A **detailed error report** can help you identify the cause of the issue.

Setting up XAMPP

A common source of error connected with Apache is **blocked ports**. If you’re using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it’s likely that the Tomcat port is being blocked, meaning the web server can’t be started. There are three ways to solve this issue:

- **Change the conflicting port:** Let’s assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype’s port settings. To do this, open the program and navigate via ‘Actions’, ‘Options’, and ‘Advanced’, until you reach the ‘Connections’ menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.

- **Change the XAMPP module port settings**: Click the Config button for the module in question and open the files *httpd.conf* and *httpd-ssl.conf*. Replace port number 80 in *httpd.conf* and port number 443 in *httpd-ssl.conf* with any free ports, before saving the file data. Now click on the general Config button on the right-hand side and select ‘Services and Ports Settings’. Customize the ports for the module server to reflect the changes in the *conf* files.

- **End the conflicting program:** The simplest way to avoid port conflicts in the short term is to end the conflicting program (Skype in this case). If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved.

Modules that can’t be started will be shown in red. The user will also receive an error report to help solve the problem

Module administration

You have an ‘Admin’ option located on the Control Panel for every module in your XAMPP.

- Click on the Admin button of your Apache server to go to the web address of your web server. The Control Panel will now start in your standard browser, and you’ll be led to the **dashboard of your XAMPP’s local host**. The dashboard features numerous links to websites for useful information as well as the open source project BitNami, which offers you many different applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can reach the dashboard through *localhost/dashboard/*.

By clicking on the 'admin' button of the Apache module, the user will be redirected to the local dashboard of XAMPP

- You can use the Admin button of your database module to open **phpMyAdmin**. Here, you can manage the databases of your web projects that you’re testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via localhost*/phpmyadmin/*.

The web project’s databases are managed by the user in php MyAdmin (accessible via the 'Admin' button in the database module)

7.2 CODE

```
<%@page import="java.sql.*" %>

<%@include file="connect.jsp" %>



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />
<script type="text/javascript" src="js/cufon-yui.js"></script>
<script type="text/javascript" src="js/cufon-aller.js"></script>
<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
<script type="text/javascript" src="js/script.js"></script>
<script type="text/javascript" src="js/coin-slider.min.js"></script>
<style type="text/css">
<!--
.style1 {font-size: 14px}
.style2 {font-size: 24px}
.style3 {
      color: #FF0000;
      font-weight: bold;
}
.style4 {font-size: 18px}
-->
</style>
</head>
<body>
<div class="main">
  <div class="header">
```

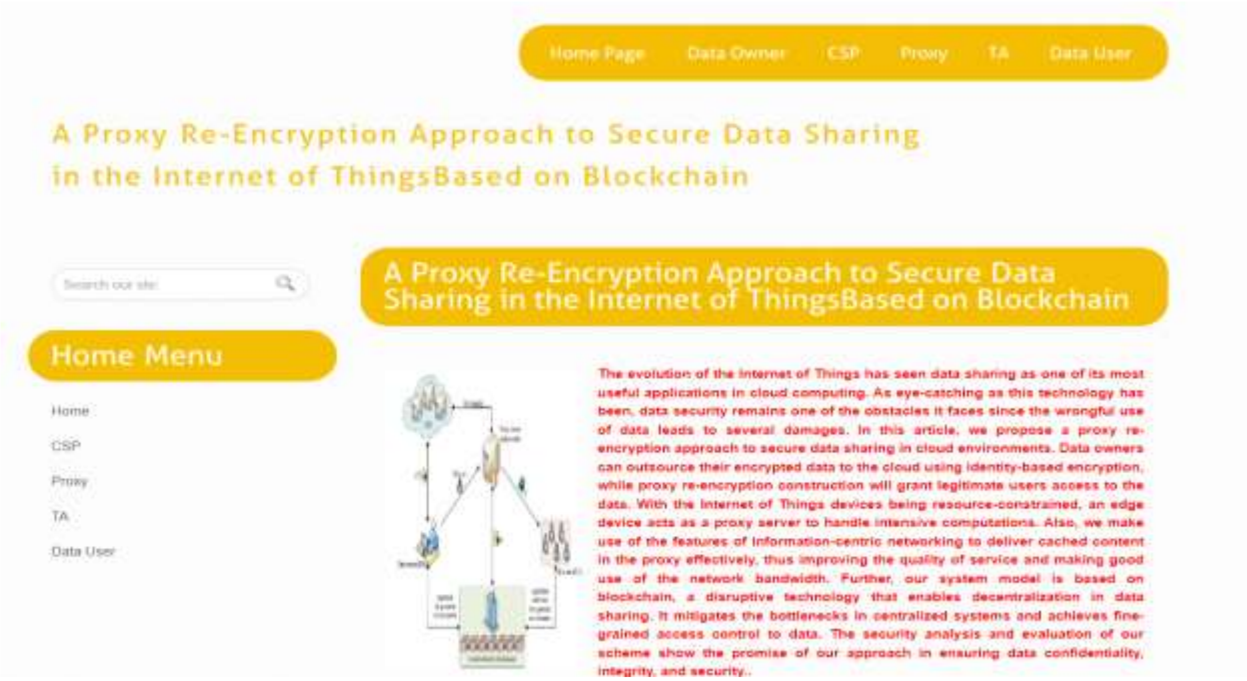
```
<div class="header_resize">

  <div class="menu_nav">

    <ul>
```

7.3 SCREAN SHORTS

7.3.1 Home page:

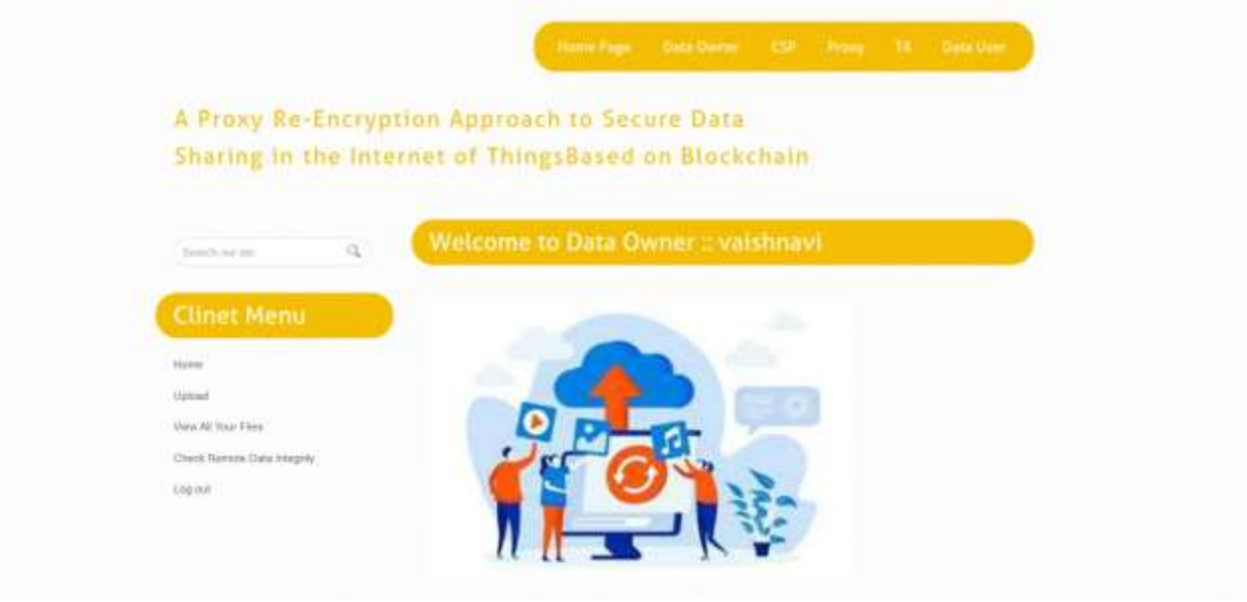


A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

The evolution of the Internet of Things has seen data sharing as one of its most useful applications in cloud computing. As eye-catching as this technology has been, data security remains one of the obstacles it faces since the wrongful use of data leads to several damages. In this article, we propose a proxy re-encryption approach to secure data sharing in cloud environments. Data owners can outsource their encrypted data to the cloud using identity-based encryption, while proxy re-encryption construction will grant legitimate users access to the data. With the Internet of Things devices being resource-constrained, an edge device acts as a proxy server to handle intensive computations. Also, we make use of the features of information-centric networking to deliver cached content in the proxy effectively, thus improving the quality of service and making good use of the network bandwidth. Further, our system model is based on blockchain, a disruptive technology that enables decentralization in data sharing. It mitigates the bottlenecks in centralized systems and achieves fine-grained access control to data. The security analysis and evaluation of our scheme show the promise of our approach in ensuring data confidentiality, integrity, and security..

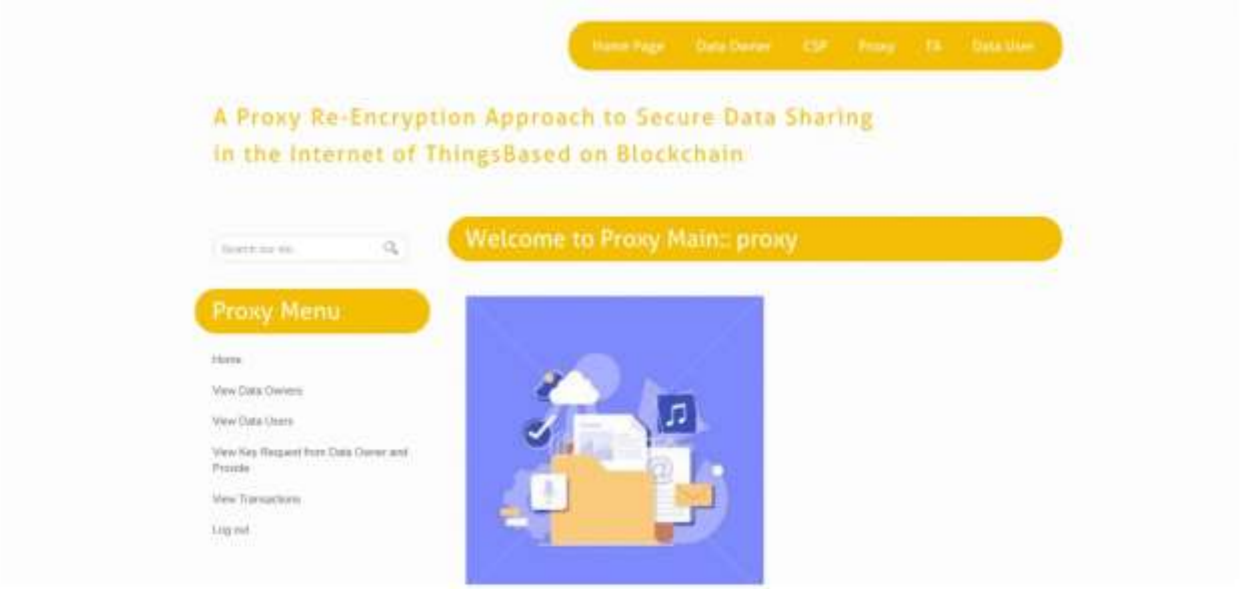
7.3.2 Owner page:



A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

Welcome to Data Owner :: vaishnavi

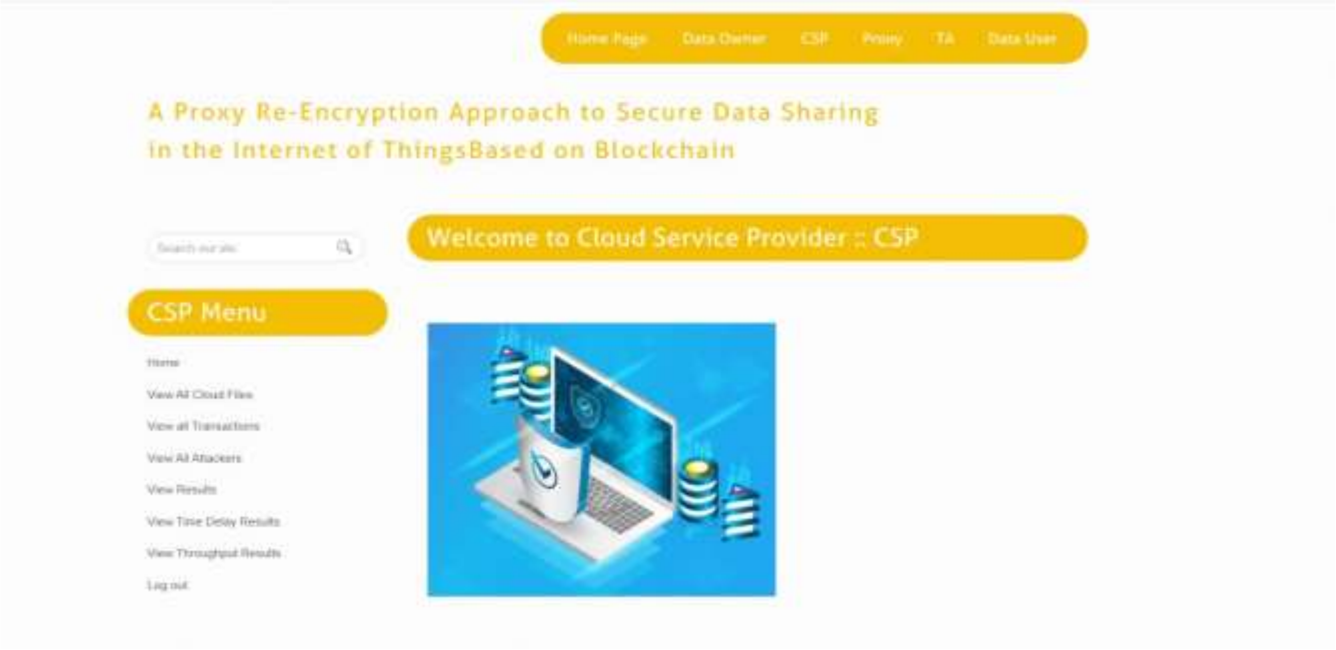
7.3.3 Proxy page:



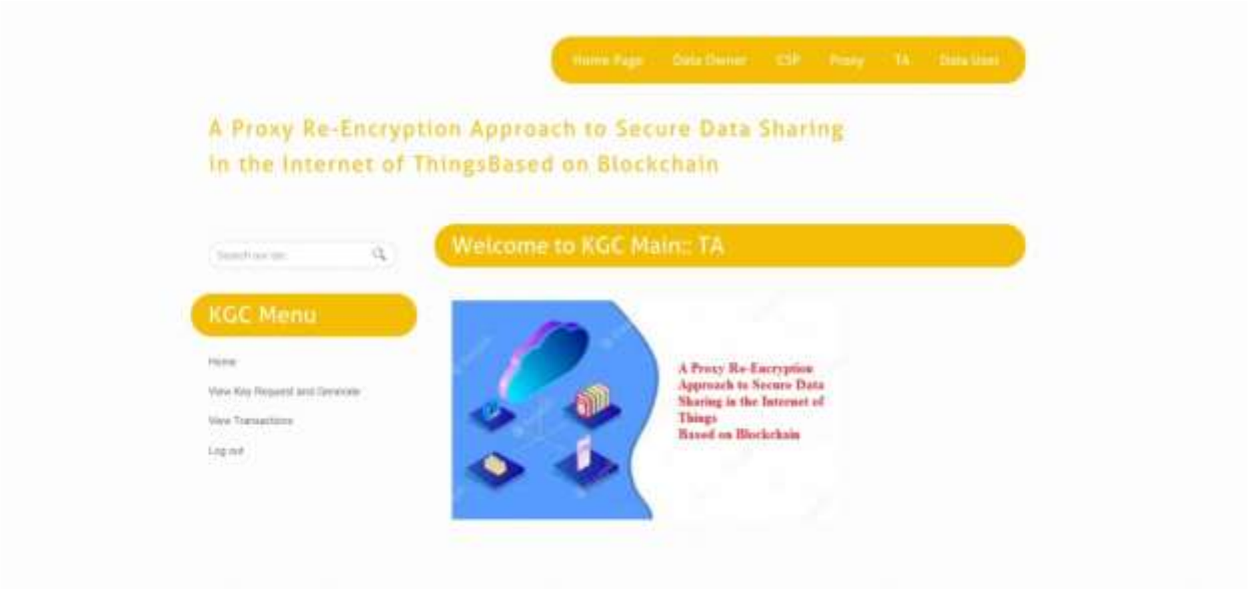
A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

Welcome to Proxy Main:: proxy

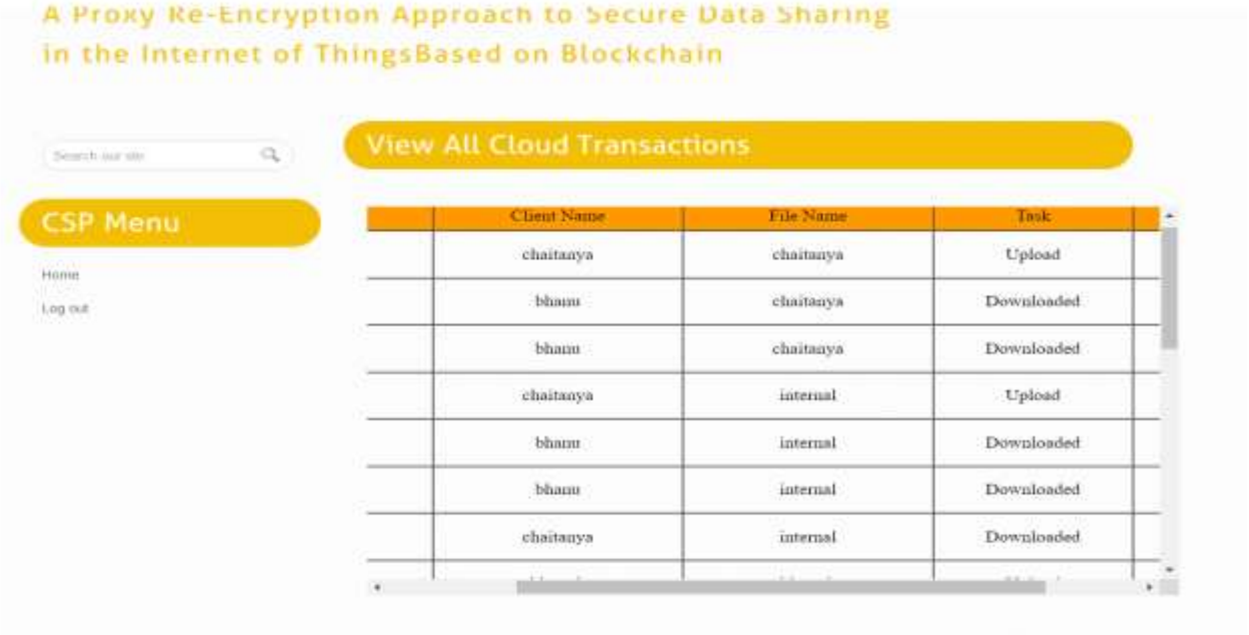
7.3.4 Cloud server page:



7.3.5 Trust authority page:



7.3.6 View Cloud Transaction Page:

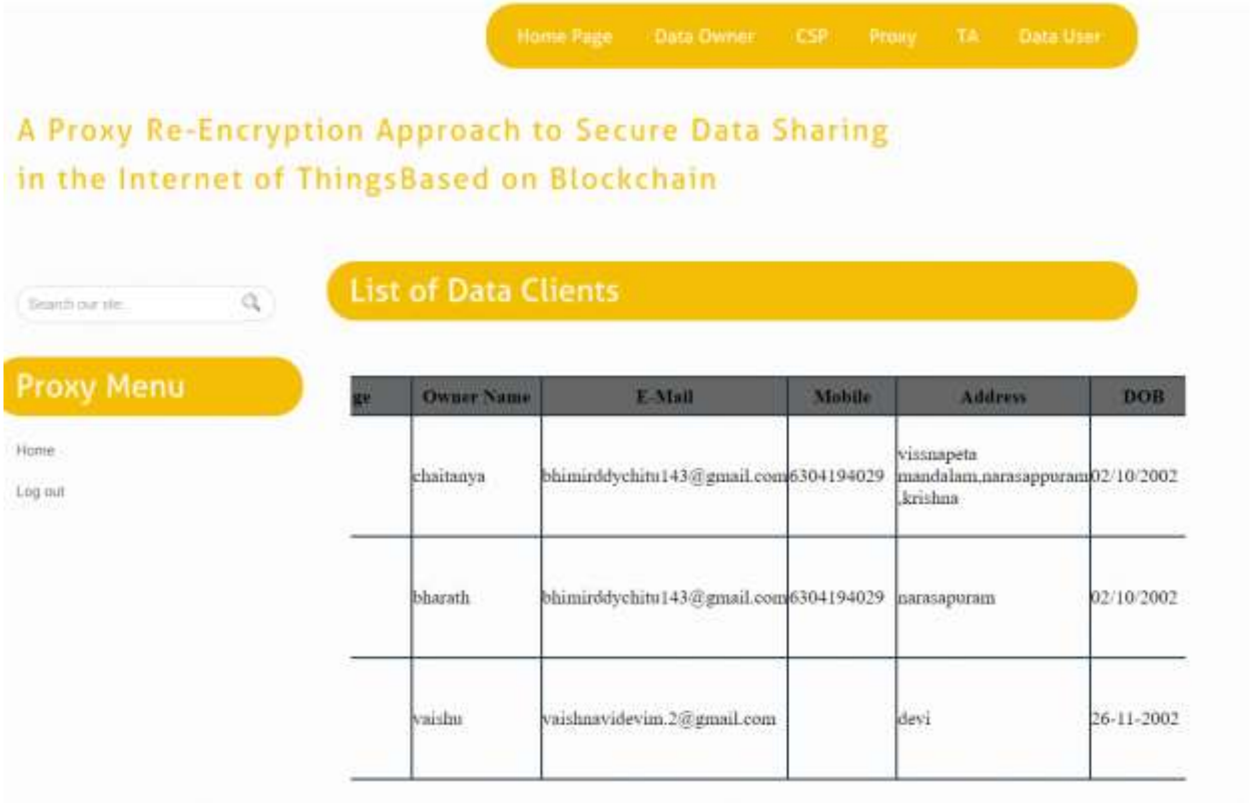


A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

View All Cloud Transactions

	Client Name	File Name	Task
	chaitanya	chaitanya	Upload
	bhanu	chaitanya	Downloaded
	bhanu	chaitanya	Downloaded
	chaitanya	internal	Upload
	bhanu	internal	Downloaded
	bhanu	internal	Downloaded
	chaitanya	internal	Downloaded

7.3.7 List of Data Clients Page:

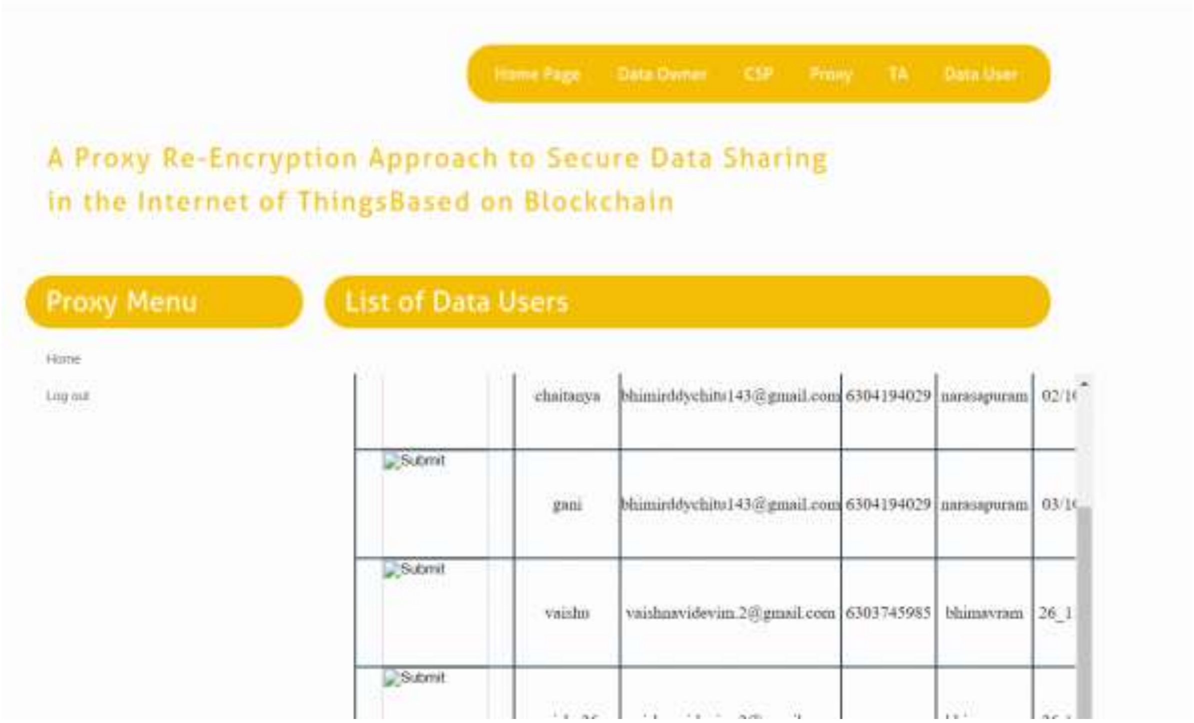


A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

List of Data Clients

ge	Owner Name	E-Mail	Mobile	Address	DOB
	chaitanya	bhimirddychitu143@gmail.com	6304194029	vissnapeta mandalam,narasappuram ,krishna	02/10/2002
	bharath	bhimirddychitu143@gmail.com	6304194029	narasapuram	02/10/2002
	vaishu	vaishnavidevim.2@gmail.com		devi	26-11-2002

7.3.8 list of data users page:

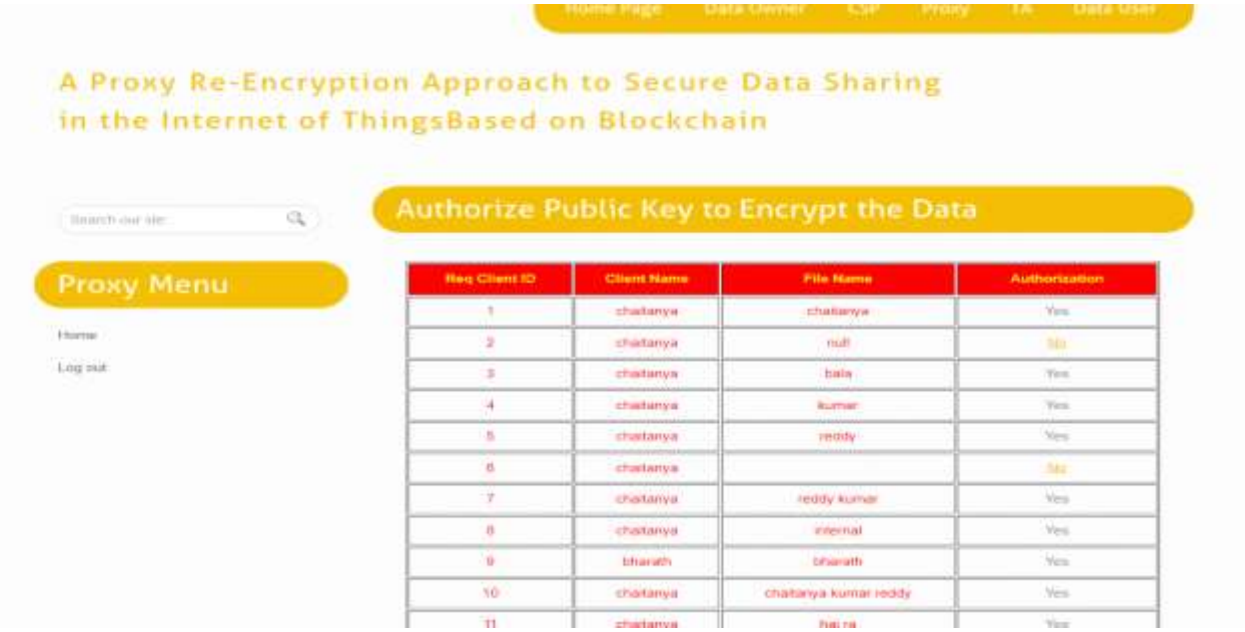


A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

List of Data Users

	chaitanya	bhimirddychitu143@gmail.com	6304194029	narasapuram	02/1
Submit	gani	bhimirddychitu143@gmail.com	6304194029	narasapuram	03/1
Submit	vaishu	vaishnavidevim.2@gmail.com	6303745985	bhimavram	26_1
Submit					

7.3.9 Key To Encrypt Data Page:



A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of ThingsBased on Blockchain

Authorize Public Key to Encrypt the Data

Req Client ID	Client Name	File Name	Authorization
1	chaitanya	chaitanya	Yes
2	chaitanya	null	No
3	chaitanya	bala	Yes
4	chaitanya	kumar	Yes
5	chaitanya	reddy	Yes
6	chaitanya		No
7	chaitanya	reddy kumar	Yes
8	chaitanya	internal	Yes
9	bharath	bharath	Yes
10	chaitanya	chaitanya kumar reddy	Yes
11	chaitanya	hai ra	Yes

8. CONCLUSION AND FUTURE ENHANCEMENT

The emergence of the IOT has made data sharing one of its most prominent applications. To guarantee data confidentiality, integrity, and privacy, we propose a secure identity-based PRE data-sharing scheme in a cloud computing environment. Secure data sharing is realized with IBPRE technique, which allows the data owners to store their encrypted data in the cloud and share them with legitimate users efficiently. Due to resource constraints, an edge device serves as the proxy to handle the intensive computations. The scheme also incorporates the features of ICN to proficiently deliver cached content, thereby improving the quality of service and making great use of the network bandwidth. Then, we present a block chain-based system model that allows for flexible authorization on encrypted data. Fine grained access control is achieved, and it can help data owners achieve privacy preservation in an adequate way. The analysis and results of the proposed model show how efficient our scheme is, compared to existing schemes

9. BIBLIOGRAPHY

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” IEEE Commun. Surveys Tut., vol. 17, no. 4, pp. 2347–2376, Oct./Dec. 2015.

[2] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., Springer, May 1998, pp. 127–144.

[3] A. Shamir, “Identity-based cryptosystems and signature schemes,” in Proc. Workshop Theory Appl. Cryptographic Techn., Springer, Aug. 1984, pp. 47–53.

[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., Springer, May 2004, pp. 506–522.

[5] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, “Building an encrypted and searchable audit log,” in NDSS, vol. 4. Citeseer, Feb. 2004, pp. 5–6.

[6] D. Balfanz et al., “Secret handshakes from pairing-based key agreements,” in Proc. IEEE, Symp. Secur. Privacy, 2003, pp. 180–196.

[7] R. Canetti, S. Halevi, and J. Katz, “Chosen-ciphertext security from identity-based encryption,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., Springer, 2004, pp. 207–222.

[8] T. Koponen et al., “A data-oriented (and beyond) network architecture,” in Proc. Conf. Appl., Techn., Architectures, Protoc. Comput. Commun., Aug. 2007, pp. 181–192.

[9] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos, “Developing information networking further: From PSIRP to pursuit,” in Proc. Int. Conf. Broadband Commun., Netw. Syst., Springer, Oct. 2010, pp. 1–13.

[10] C. Dannewitz, J. Golic, B. Ohlman, and B. Ahlgren, “Secure naming for a network of information,” in Proc. INFOCOM IEEE Conf. Comput. Commun. Workshops,2010, pp. 1–6.

[11] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, “A routing scheme for content-based networking,” in Proc. IEEE INFOCOM 2004, vol. 2, 2004, pp. 918–928.

[12] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information-centric networks,” in Proc. 2nd ed. ICN Workshop Inform.- Centric Netw., Aug. 2012, pp. 55–60.

[13] Y. Sun et al., “Trace-driven analysis of ICN caching algorithms on videoon-demand workloads,” in Proc. 10th ACM Int. Conf. Emerging Netw. Exp. Technol., Dec. 2014, pp. 363–376.

[14] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, vol. 4. Bitcoin.org, 2008. Available: https://bitcoin. org/bitcoin. pdf

[15] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–9.

[16] N. Park, “Secure data access control scheme using type-based reencryption in cloud environment,” in Semantic Methods Knowledge Management and Communications. Berlin, Germany: Springer, 2011, pp. 319– 327

10.WEB LINKS

https://www.youtube.com/watch?v=9tZ23OEUAU4

https://en.wikipedia.org/wiki/Proxy_re-encryption#:~:text=Proxy%20re%2Dencryption%20(PRE),may%20be%20decrypted%20by%20another.

https://ieeexplore.ieee.org/document/9442931