

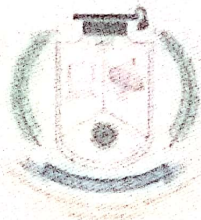
DEPLOYMENT OF BLOCKCHAIN IN CLOUD COMPUTING

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

By

SANKRANTHI V S A SANTOSH(40111141)

HARI KRISHNA SAI NIDADAVOLU(40110811)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL-2024

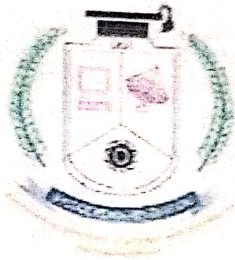
DEPLOYMENT OF BLOCKCHAIN IN CLOUD COMPUTING

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

By

SANKRANTHI V S A SANTOSH(40111141)

HARI KRISHNA SAI NIDADAVOLU(40110811)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Category-1 University by UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119

APRIL-2024





SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SANKRANTHI V S A SANTOSH(40111141)** and **HARI KRISHNA SAI NIDADAVOLU(40110811)** who carried out the Project work entitled **DEPLOYMENT OF BLOCK CHAIN IN CLOUD COMPUTING** under my supervision from November 2023 to April 2024.

Internal Guide

Dr. R. RAJALAKSHMI M.E., Ph.D.

Head of the Department

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Ph.D.

Submitted for Viva voce Examination held on

Josephar Nagar, Rajiv Gandhi Salai,
Chennai - 600 093.

12/04/24

Internal Examiner

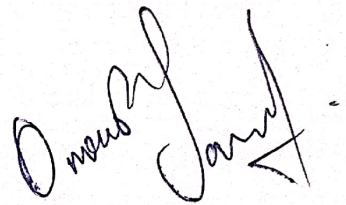
External Examiner

DECLARATION

I, **SANKRANTHI V S A SANTOSH** (Reg.No- 40111141), hereby declare that the Project Report entitled **DEPLOYMENT OF BLOCK CHAIN IN CLOUD COMPUTING** done by me under the guidance of **Dr.R. RAJALAKSHMI M.E.,Ph.D** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 12th April 2024

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R.RAJALAKSHMI M.E.,Ph.D.**,for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Nowadays, large amounts of data are stored with cloud service providers. Third-party auditors (TPAs), with the help of cryptography, are often used to verify this data. Cloud Data Auditing Techniques with a Focus on Privacy and Security. It aims to provide a resource based on-demand. It avoids online usage burden of accessing data through internet. Cloud storage supports to maintain data securely in cloud. Cloud is interconnected with group of computers, which is used to store information and run their applications in cloud platform. Through cloud computing, we can access any file, document of user from anywhere in the world. Mainly, cloud can be used for cost savings, high scalability and large storage space. But a major issues in cloud computing is security. It is an era of digitalization that includes all types of digital activities and information. In this digitalization period, if we want to convey signals or data or any digital favors, Cloud Computing has moved out of and become visible as an unbeatable technology. It forecasts divisional computing assets as alternative configuring regional servers to be head of, a stockpile of data kept for use as needed, and procedural data. The requirement for cloud computing is increasing day by day. Cloud computing is not more secure to handle entire data of this world or digital era. There are adequate security issues that come out of this increasing technology that creates a hurdle for improvements. All the information and prime data still convey from one end to another ruin through a global medium called the internet. But the data security is at risk always. For security purposes, we have to keep up the apparatus for preserving data such as privacy or confidentiality, dispensation, accuracy, answerability, and access control. There is one technology that is trending now a day called a blockchain. Blockchain makes cloud computing more secure and more powerful technology. The condition of being free from danger or threat, that state comes from a blockchain in cloud computing. All complications in certainty can solve by blockchain technology. A blockchain can increase reliability and privacy from end to end of undeniable ration disperse registry or logs on cloud confluence.

TABLE OF CONTENTS

CHAPTER No	TITLE	Page No
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 Overview of Cloud Computing	2
	1.2 Cloud Computing Models	2
	1.3 Understanding Public and Private Cloud	4
	1.4 Cloud Computing Benefits	5
	1.5 Cloud Computing Challenges	6
2	LITERATURE SURVEY	7
	2.1 Cloud Data Auditing Techniques	7
	2.2 A Survey on Auditing Techniques	7
	2.3 Auditing in Cloud Computing Solutions	8
	2.4 Cloud Security Auditing Challenges	8
	2.5 Dynamic-Hash Table Based Public Auditing	8
	2.6 Open Problems in Existing System	9
3	REQUIREMENT ANALYSIS	11
	3.1 Data Recovery and Availability	11
	3.2 Management Capabilities	11

3.3	Requirement and Analysis	11
3.4	Hardware and Software Requirements	13
4	DESCRIPTION OF PROPOSED SYSTEM	14
4.1	Existing System	14
4.2	Proposed System	15
4.3	System Architecture	16
5	IMPLEMENTAION DETAILS	17
5.1	Modules	17
5.2	UMLS(Unified Modelling Language)	18
5.3	Auditing for Secure Data Storage in cloud	19
5.4	Software Environment	26
5.5	Collection Framework	29
5.6	Testing Process	30
5.7	Test Strategy and Approach	33
6	RESULTS AND DISCUSSION	35
6.1	Presentation of Findings	35
6.2	Interpretation of Results	36
7	CONCLUSION	37
7.1	Recommendations and Future Research	37
	REFERENCES	39
	APPENDIX	43
	A. SOURCE CODE	43
	B. SCREENSHOTS	54
	C. RESEARCH PAPER	59
	D. CERTIFICATE	65

LIST OF FIGURES

CHAPTER NO	FIGURE NAME	PAGE NO
4.3	System Architecture Diagram	16
4.4	Data Flow Diagram	16
5.3.1	Class Diagram Auditing for secure data	20
5.3.2	UseCase Diagram	21
5.3.3	Sequence Diagram	22
5.3.4	Component Diagram	23
5.3.5	Activity Diagram	24
5.3.6	ER-Diagram	25
5.4.2.1	Java Platform Independent	28
5.5.1	List of Java Collections	30

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CSP	Cloud Service Provide
IaaS	Infrastructure as a Service
P2P	Peer to Peer Network
SaaS	Software as a Service

CHAPTER 1

INTRODUCTION

Storing large amounts of data with cloud service providers (CSPs) raises concerns about data protection. Data integrity and privacy can be lost because of the physical movement of data from one place to another by the cloud administrator, malware, dishonest cloud providers, or other malicious users who might distort the data.¹ Hence, saved data corrections must be verified at regular intervals.

Nowadays, with the help of cryptography, verification of remote (cloud) data is performed by third-party auditors (TPAs). TPAs are also appropriate for public auditing, offering auditing services with more powerful computational and communication abilities than regular users.³ In public auditing, a TPA is designated to check the correctness of cloud data without retrieving the entire dataset from the CSP. However, most auditing schemes don't protect user data from TPAs; hence, the integrity and privacy of user data are lost. Our research focuses on cryptographic algorithms for cloud data auditing and the integrity and privacy issues that these algorithms face. Many approaches have been proposed in the literature to protect integrity and privacy; they're generally classified according to data's various states: static, dynamic, multi owner, multiuser, and so on.

We provide a systematic guide to the current literature regarding comprehensive methodologies. We not only identify and categorize the different approaches to cloud data integrity and privacy but also compare and analyze their relative merits. For example, our research lists the strengths and weaknesses of earlier work on cloud auditing, which will enable researchers to design new methods. Although related topics such as providing security to the cloud are beyond this article's scope, cloud data auditing requires explicit attention.

1.1 OVERVIEW OF CLOUD COMPUTING:

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly.

Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment.

The idea of cloud computing is based on a very fundamental principal of „reusability of IT capabilities'. The difference that cloud computing brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries

1.2 Cloud Computing Models

Cloud Providers offer services that can be grouped into three categories.

1.2.1 Software as a Service (SaaS):

In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers' side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained [1]. One of the key advantages of SaaS is its ease of deployment. Since applications are hosted and managed by the service provider, users can access them instantly without the need for lengthy installation processes or

infrastructure setup. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

1.2.2 Platform as a Service (Paas):

Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySQL and PHP), restricted J2EE, Ruby etc. Google's App Engine, Force.com, etc are some of the popular PaaS examples.

1.2.3 Infrastructure as a Service (IaaS)

the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. One of the primary benefits of IaaS is its scalability. Users can easily provision and scale virtual resources such as virtual machines, storage, and network bandwidth to meet changing workload requirements [2]. And another advantage is Cost-effectiveness is another key advantage of IaaS. Instead of purchasing and maintaining hardware infrastructure, businesses pay only for the resources. Whether experiencing sudden spikes in traffic or expanding business operations, organizations can rapidly allocate additional resources to support their needs, ensuring optimal performance and availability. Instead of purchasing and maintaining hardware infrastructure, businesses pay only for the resources. Whether experiencing Some common examples are Amazon, GoGrid, 3 Tera, etc.

1.3 Understanding Public and Private Clouds

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds

1.3.1 Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, "Pay-as-you-go" model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

1.3.2 Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. There are two variations to a private cloud:

1.3.3 On-premise Private Cloud:

On-premise private clouds, also known as internal clouds are hosted within one's own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability. IT departments would also need to incur the capital and operational costs for the physical resources [3]. This is best suited for applications which require complete control and configurability of the infrastructure and security.

1.3.4 Externally hosted Private Cloud:

This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don't prefer a public cloud due to sharing of physical resources.

1.3.5 Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

1.4 Cloud Computing Benefits

Enterprises would need to align their applications, so as to exploit the architecture models that Cloud Computing offers. Some of the typical benefits are listed below:

1) Reduced Cost

There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance. Initial expense and recurring expenses are much lower than traditional computing.

2) Increased Storage

With the massive Infrastructure that is offered by Cloud providers today, storage & maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

3) Flexibility

This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

1.5 Cloud Computing Challenges:

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

- **Data Protection**

Data Security is a crucial element that warrants scrutiny. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumers. In many instances, the actual storage location is not disclosed, adding onto the security concerns of enterprises. In the existing models, firewalls across data centers (owned by enterprises) protect this sensitive information. In the cloud model, Service providers are responsible for maintaining data security and enterprises would have to rely on them. One of the primary challenges enterprises face is the reluctance to fully trust vendors' assurances regarding data security. Companies fear the potential consequences of data breaches, including loss of competitive advantage, reputational damage, and legal liabilities. Moreover, the confidentiality of consumer data is paramount, and any compromise in data security can lead to significant financial and regulatory repercussions. As a result, enterprises are cautious about entrusting their sensitive information to external parties without robust guarantees of data protection. In many instances, the lack of transparency regarding the actual storage location of data exacerbates these security concerns. Enterprises may not have visibility into where their data is physically housed, raising questions about data residency, sovereignty, and compliance with regulatory requirements. This opacity can undermine enterprises' confidence in the security measures implemented by vendors and heighten their apprehensions about relinquishing control over their data. Traditionally, enterprises have relied on internal firewalls and security measures to safeguard their data within on-premises data centers. However, the shift towards cloud computing models introduces a new paradigm, wherein service providers assume greater responsibility for maintaining data security. This transition necessitates a shift in mindset, wherein enterprises must place trust in their chosen cloud service providers [4].

CHAPTER 2

LITERATURE SURVEY

2.1 Cloud Data Auditing Techniques with a Focus on Privacy and Security:

Nowadays, large amounts of data are stored with cloud service providers. Third-party auditors (TPAs), with the help of cryptography, are often used to verify this data. However, most auditing schemes don't protect cloud user data from TPAs. A review of the state of the art and research in cloud data auditing techniques highlights integrity and privacy challenges, current solutions, and future research directions.

2.2 A survey on auditing techniques used for preserving privacy of data stored on cloud:

Providing security to the stored data on the cloud is one of the important challenges in cloud computing. Encrypted data which is stored on the cloud may be viewed or modified by the cloud service provider. To overcome this problem many techniques have been developed but, those cannot guarantee accurately about the security of the stored data. These modifications of the data by the service provider or by others should also be known to the data owner [5]. For such purpose, data tagging technique can be used to audit the data. Auditing is done by using Third Party Auditor (TPA). TPA stores data information of the data owner and challenges to the cloud server, depending upon the data owner request. With the help of such mechanism, TPA can convince both, data owner and cloud server.

2.3 Auditing in Cloud Computing Solutions with OpenStack:

This presentation will walk through how auditing works in a Cloud environment. We will touch upon things like Cloud Auditing Data standard (CADF), the auditing challenges in a distributed cloud platform like OpenStack and how they are overcome using by CADF.

2.4 Cloud Security Auditing: Challenges and Emerging Approaches:

IT auditors collect information on an organization's information systems, practices, and operations and critically analyze the information for improvement. One of the primary goals of an IT audit is to determine if the information system and its maintainers are meeting both the legal expectations of protecting customer data and the company standards of achieving financial success against various security threats. These goals are still relevant in the newly emerging cloud computing model of business, but they need customization [6]. There are clear differences between cloud and traditional IT security auditing. In this article, the authors explore potential challenges unique to cloud security auditing; examine additional challenges specific to particular cloud computing domains such as banking, medical, and government sectors; and present emerging cloud-specific security auditing approaches and provide critical analysis.

2.5 Dynamic-Hash-Table Based Public Auditing for Secure CloudStorage:

Cloud storage is an increasingly popular application of cloud computing, which can provide on-demand outsourcing data services for both organizations and individuals. However, users may not fully trust the cloud service providers (CSPs) in that it is difficult to determine whether the CSPs meet their legal expectations for data security. Therefore, it is critical to develop efficient auditing techniques to strengthen data owners' trust and

confidence in cloud storage. In this paper, we present a novel public auditing scheme for secure cloud storage based on dynamic hash table (DHT), which is a new two-dimensional data structure located at a third party auditor (TPA) to record the data property information for dynamic auditing. Differing from the existing works, the proposed scheme migrates the authorized information from the CSP to the TPA, and thereby significantly reduces the computational cost and communication overhead [7]. Meanwhile, exploiting the structural advantages of the DHT, our scheme can also achieve higher updating efficiency than the state-of-the-art schemes. In addition, we extend our scheme to support privacy preservation by combining the homomorphic authenticator based on the public key with the random masking generated by the TPA, and achieve batch auditing by employing the aggregate BLS signature technique. We formally prove the security of the proposed scheme, and evaluate the auditing performance by detailed experiments and comparisons with the existing ones. The results demonstrate that the proposed scheme can effectively achieve secure auditing for cloud storage, and outperforms the previous schemes in computation complexity, storage costs and communication overhead.

2.1 OPEN PROBLEMS IN EXISTING SYSTEM

Abuse and Nefarious Use of Cloud Computing

IaaS providers offer their customers the illusion of unlimited compute, network, and storage capacity often coupled with a 'frictionless' registration process where anyone with a valid credit card can register and immediately begin using cloud services. Some providers even offer free limited trial periods. By abusing the relative anonymity behind these registration and usage models, spammers, malicious code authors, and other criminals have been able to conduct their activities with relative impunity. PaaS providers have traditionally suffered most from this kind of attacks; however, recent evidence shows that hackers have begun to target IaaS vendors as well.

Future areas of concern include

password and key cracking, DDOS, launching dynamic attack points, hosting malicious data, botnet command and control, building rainbow tables, and CAPTCHA solving farms.

- **Insecure Interfaces and APIs**

Cloud computing providers expose a set of software interfaces or APIs that customers use to manage and interact with cloud services [8]. Provisioning, management, orchestration, and monitoring are all performed using these interfaces. The security and availability of general cloud services is dependent upon the security of these basic APIs. This introduces the complexity of the new layered API; it also increases risk, as organizations may be required to relinquish their credentials to third parties in order to enable their agency.

- **Malicious Insiders**

The threat of a malicious insider is well-known to most organizations. This threat is amplified for consumers of cloud services by the convergence of IT services and customers under a single management domain, combined with a general lack of transparency into provider process and procedure

- **Shared Technology Issues**

IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure were not designed to offer strong isolation properties for a multi-tenant architecture.

- **Data Loss or Leakage**

There are many ways to compromise data. Deletion or alteration of records without a backup of the original content is an obvious example

- **Account or Service Hijacking**

Account or service hijacking is not new. Attack methods such as phishing, fraud, and exploitation of software vulnerabilities still achieve results

- **Unknown Risk Profile**

Security by obscurity may be low effort, but it can result in unknown exposures. It may also impair the in-depth analysis required highly controlled or regulated operational areas.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Data Recovery and Availability

All business applications have Service level agreements that are stringently followed. Operational teams play a key role in management of service level agreements and runtime governance of applications. In production environments, operational teams support appropriate clustering and Fail over Data Replication System monitoring (Transactions monitoring, logs monitoring and others) Maintenance (Runtime Governance) Disaster recovery Capacity and performance management if, any of the above mentioned services is under-served by a cloud provider, the damage & impact could be severe.

3.2 Management Capabilities

Despite there being multiple cloud providers, the management of platform and infrastructure is still in its infancy. Features like „Auto-scaling“ for example, are a crucial requirement for many enterprises. There is huge potential to improve on the scalability and load balancing features provided today.

3.2.1 Regulatory and Compliance Restrictions

In some of the European countries, Government regulations do not allow customer's personal information and other sensitive information to be physically located outside the state or country. In order to meet such requirements, cloud providers need to setup a datacenter or a storage site exclusively within the country to comply with regulations. Having such an infrastructure may not always be feasible and is a big challenge for cloud providers.

3.3 REQUIREMENT & ANALYSIS

The software requirement specification gives the system specification in which process requirements are presented in an easily understandable way. Thus it contains all the inputs required, processes in the system and outputs produced by the system.

Software Requirements Specification plays an important role in creating quality software solutions. Specification is basically a representation process. Requirements are represented in a manner that ultimately leads to successful software implementation.

Requirements may be specified in a variety of ways. However there are some guidelines worth following

Representation format and content should be relevant to the problem.

Information contained within the specification should be nested.

Requirement analysis enables the system engineer to specify software function and perform, indicate software's interface with the other system elements, and establish constraints that software must meet [8]. Requirement analysis allows the analyst to refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software.

The first step is to understand the user's requirement within the framework of the organization's objectives and the environment in which the system is installed. Considerations are given to the user to carry on with the work within the organization's specified objectives.

Using swings in java we will develop the proposed system. The proposed application can be implemented by taking minimum of three systems into consideration. The server is implemented in one system TPA is implemented in another system and client can be implemented from n no of systems it can be implemented on any operating system like windows or linux. The client system will store data like files images on to server through TPA. The TPA will store the metadata information of file on TPA where as server will store files as well as metadata data information about the files. Whenever the client asks for the verification of files on cloud the TPA will check for the data integrity on the server. This application demands minimum three systems should be connected within a network.

3.4 HARDWARE AND SOFTWARE REQUIREMENTS

Database	: MySQL
Operating System	: Windows95/98/2000/XP
Processor	: Pentium 4 processor
RAM	: 1 GB RAM
Hard Disk	: 80 GB Hard Disk Space

CHAPTER 4 DESCRIPTION OF PROPOSED SYSTEM

4.1 EXISTING SYSTEM

While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. Although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. There do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding their outsourced data status [9]. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture.

As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in addition to retrieving the data). In particular, users may not want to go through the

accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party.

4.2 PROPOSED SYSTEM

The proposed system can be summarized as the following three aspects:

We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol, i.e., our scheme supports an external auditor to audit user's outsourced data in the cloud without learning knowledge on the data content [10].

- 1) To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing in the Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
- 2) We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

ADVANTAGES

- Novel automatic and enforceable logging mechanism in the cloud.
- Proposed architecture is platform independent and highly decentralized, in that it does not require any dedicated authentication or storage system in place.
- Provide a certain degree of usage control for the protected data after these are delivered to the receiver.
- The results demonstrate the efficiency, scalability, and granularity of our approach. We also provide a detailed security analysis and discuss the reliability and strength of our architecture.

4.2 ARCHITECTURE:

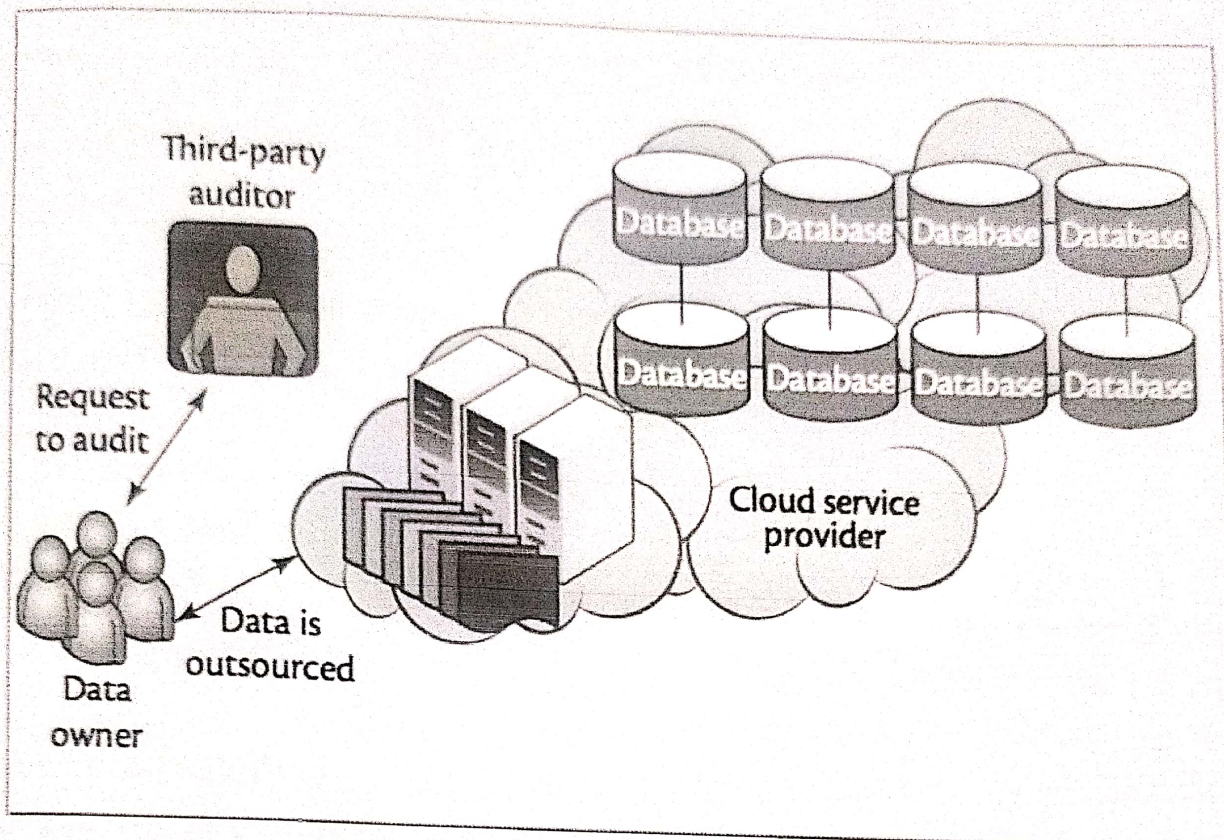


Fig 4.3 System Architecture

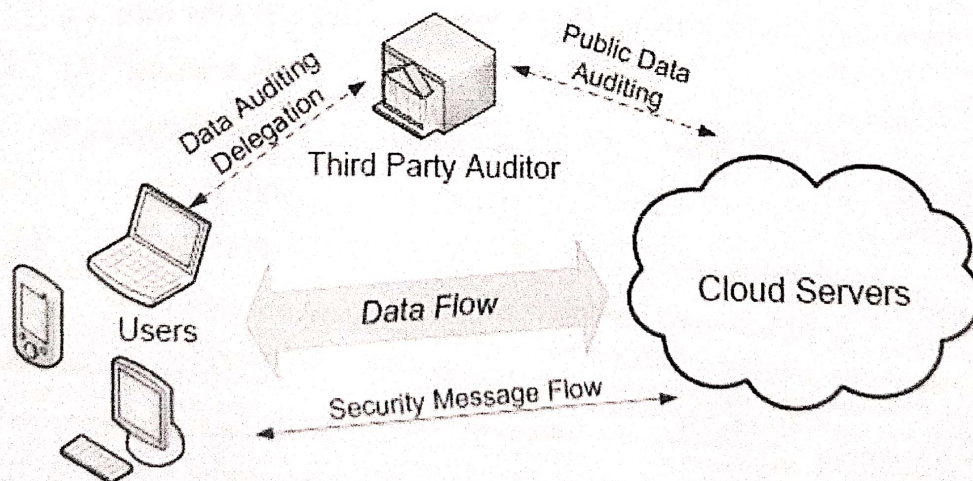


Fig 4.4 Data flow diagram

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 MODULES

The system is proposed to have the following modules:

- Admin Module
- TPA module
- User Module
- Block Verification Module
- Block Insertion Module
- Block Deletion

5.1.1 ADMIN MODULE

Admin is allowed to check which user registered and which data is stored in the cloud space area

5.1.2 TPA MODULE

TPA check that data is modified or not if modified that information send to use

5.1.3 USER MODULE

User can register and he can login with his user id and password and he can upload the data to cloud space area

5.1.4 BLOCK VERIFICATION MODULE

User can check that the uploaded file is modified by any one or not (like server area)

5.1.5 BLOCK INSERTION MODULE

In the block insertion module user can insert the new block

5.1.6 BLOCK DELETION MODULE

In the Block Deletion Module user can delete the Block.

5.2 UMLS

UNIFIED MODELING LANGUAGE

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows

5.2.1 User Model View

This view represents the system from the user's perspective.

The analysis representation describes a usage scenario from the end-users perspective

5.2.2 Structural model view

In this model the data and functionality are arrived from inside the system.

This model view models the static structures.

5.2.3 Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the usermodel and structural model view.

5.2.5 Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

5.2.4 Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the system.
- UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer ...etc., or another system like central database.

5.1 CLASS DIAGRAM AUDITING FOR SECURE DATA STORAGE IN CLOUD

This class diagram contains four classes that are Server, TPA, Client and Database. Server will perform operations like it maintains client details & session information, stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the relationship between these classes.

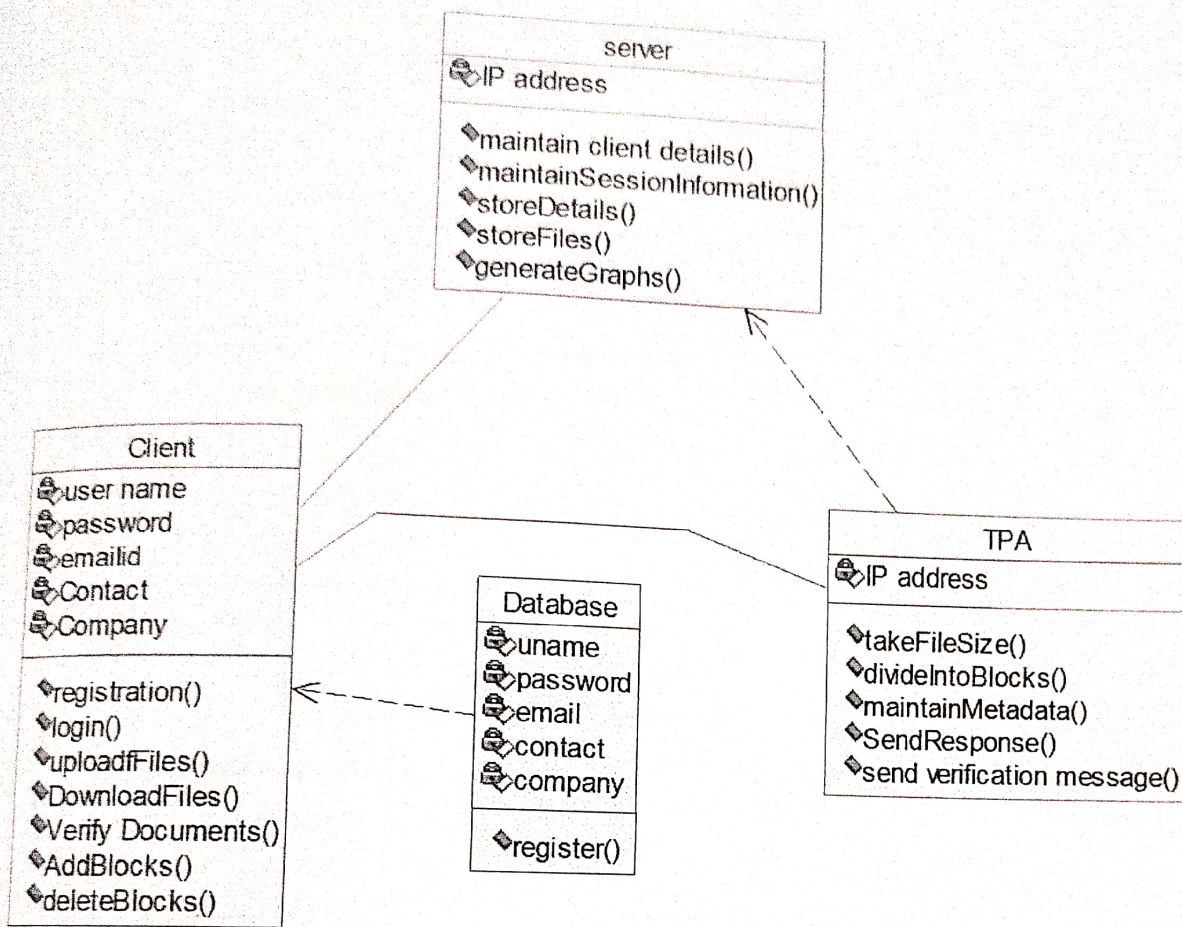


Fig 5.3.1 Class Diagram

This use case diagram contains three actors that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information, stores details & files and generate graphs [11]. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the use cases of each actor and relationship between this actors and use cases.

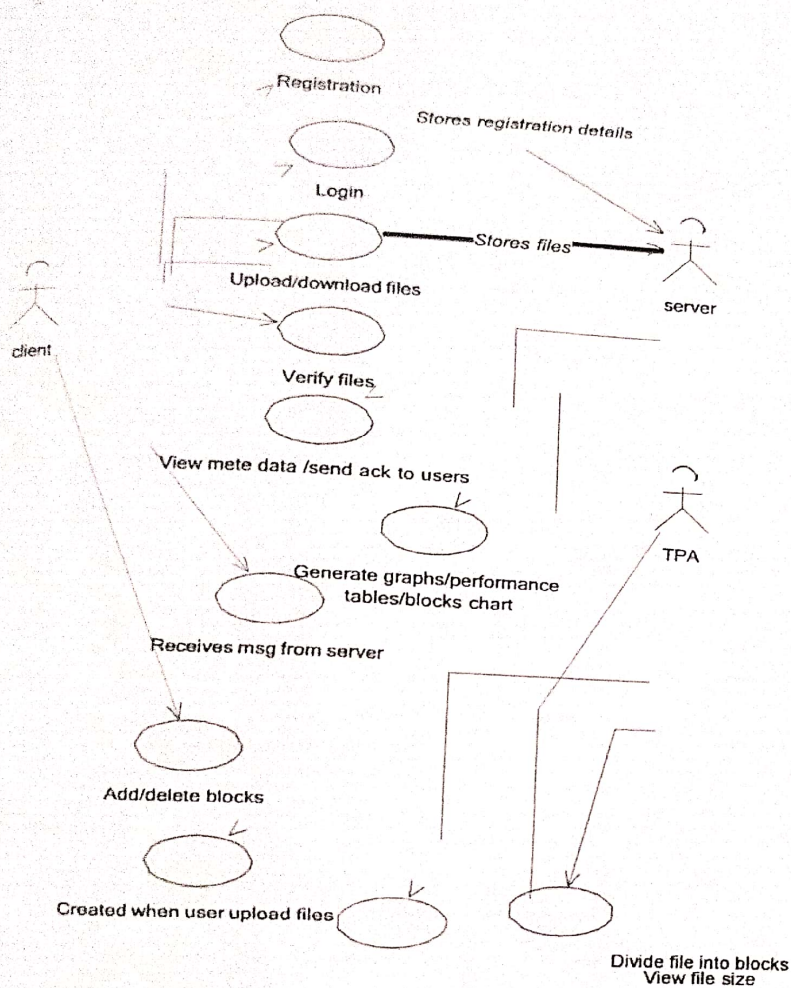


Fig 5.3.2 Usecase Diagram

5.3.3 SEQUENCE DIAGRAM:

This sequence diagram contains three objects that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information stores details& files and generate graphs. Client will perform operations like registration, login, uploadfiles, download files, verify documents, add blocks, delete blocks. TPA will perform

operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the sequence of actions performed between these objects.

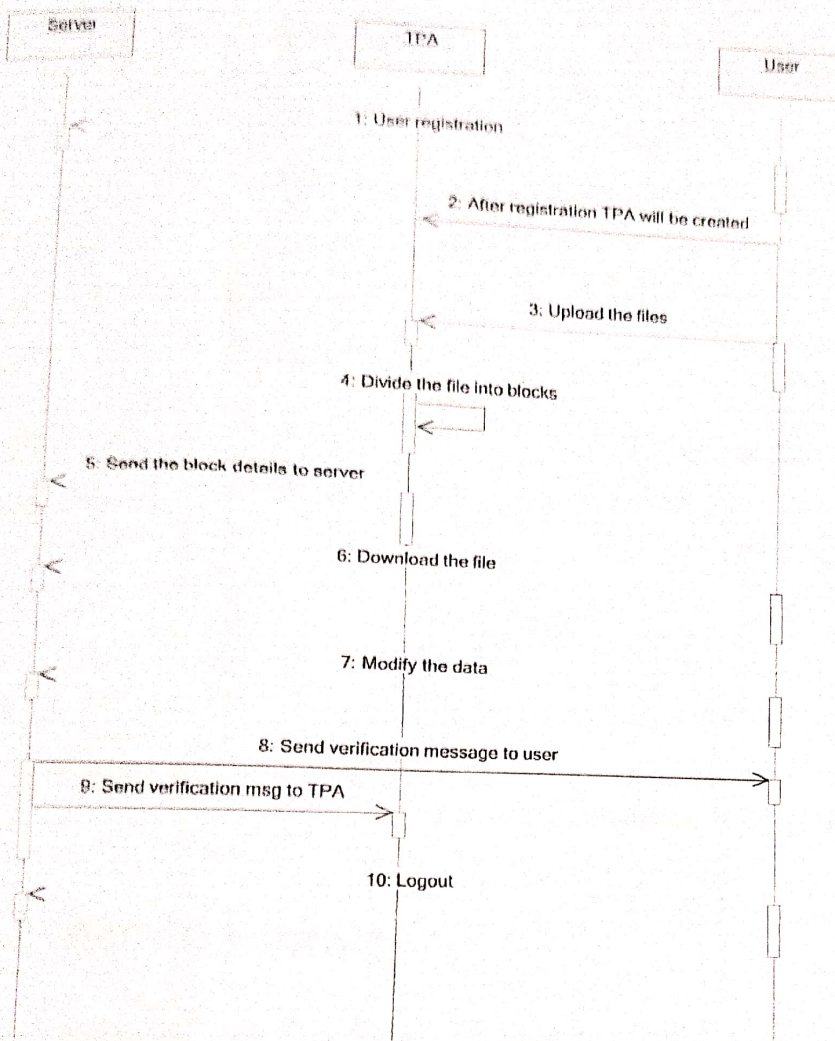


Fig 5.3.3 Sequence Diagram

5.3.4 COMPONENT DIAGRAM:

This component diagram contains three components that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information stores details & files and generate graphs. Client will perform operations like registration, login, upload files, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divided file into blocks, maintain metadata information, send

response and verification message. And this diagram shows the actions performed by these components.

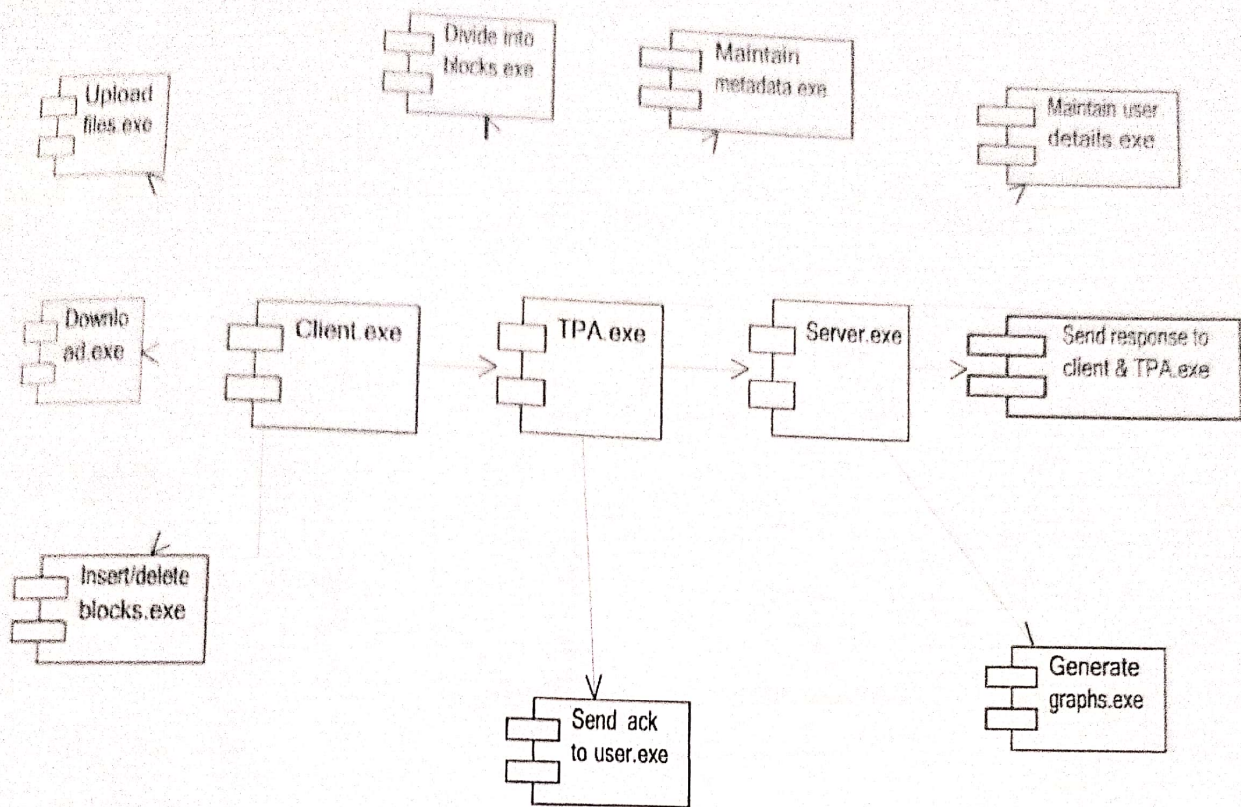


Fig 5.3.4 Component Diagram

5.3.5 ACTIVITY DIAGRAM:

This activity diagram contains three activities that are Server, TPA, Client and. This diagram shows the flow of control between these activities.

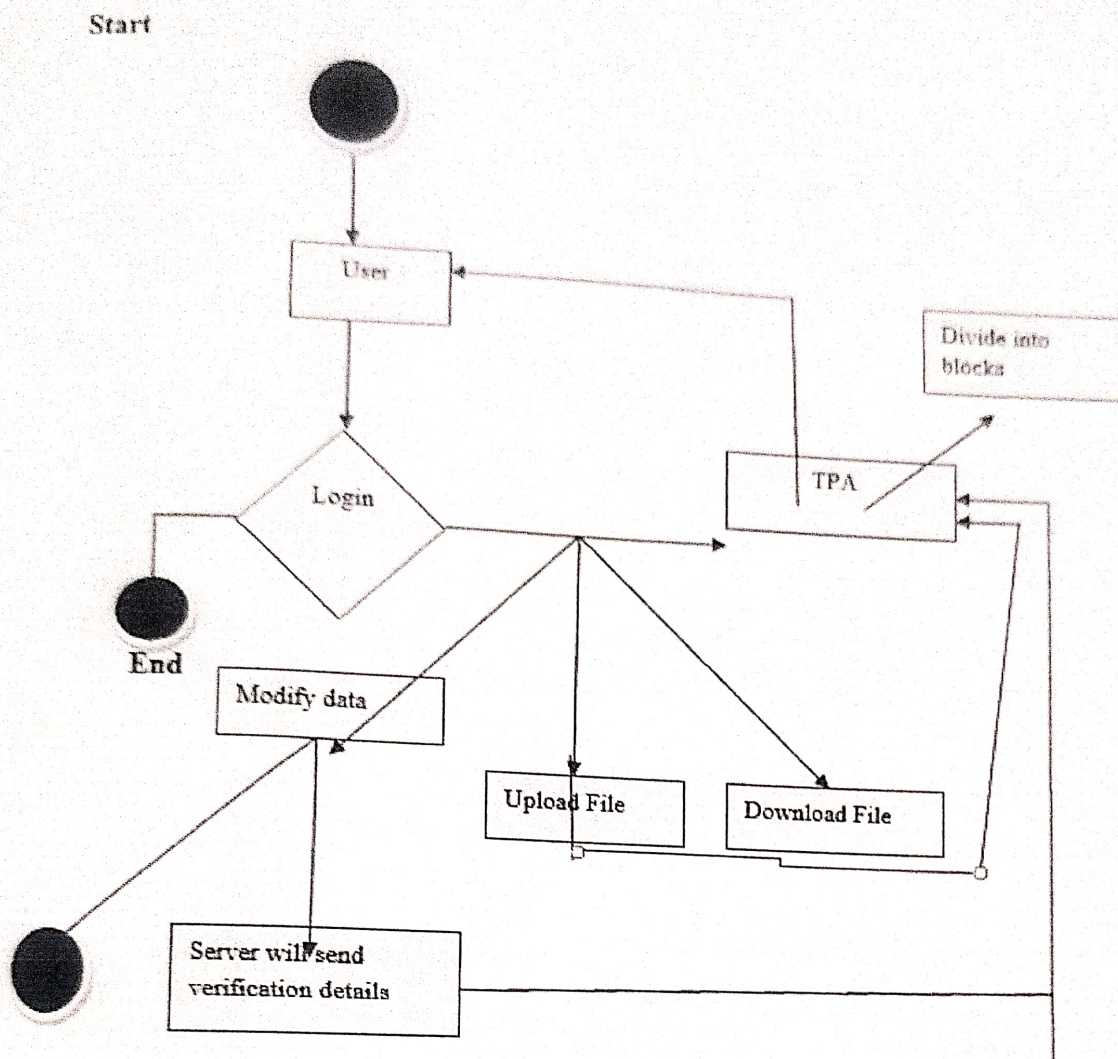


Fig 5.3.5 Activity Diagram

5.3.6 ER-DIAGRAMS

This ER-Diagram contains three entities that are Server, TPA, Client and. Server will perform operations like it maintains client details & session information stores details& files and generate graphs. Client will perform operations like registration, login, uploadfiles, download files, verify documents, add blocks, delete blocks. TPA will perform operations like take file size, divide file into blocks, maintain metadata information, send response and verification message. And this diagram shows the relationship between these entities.

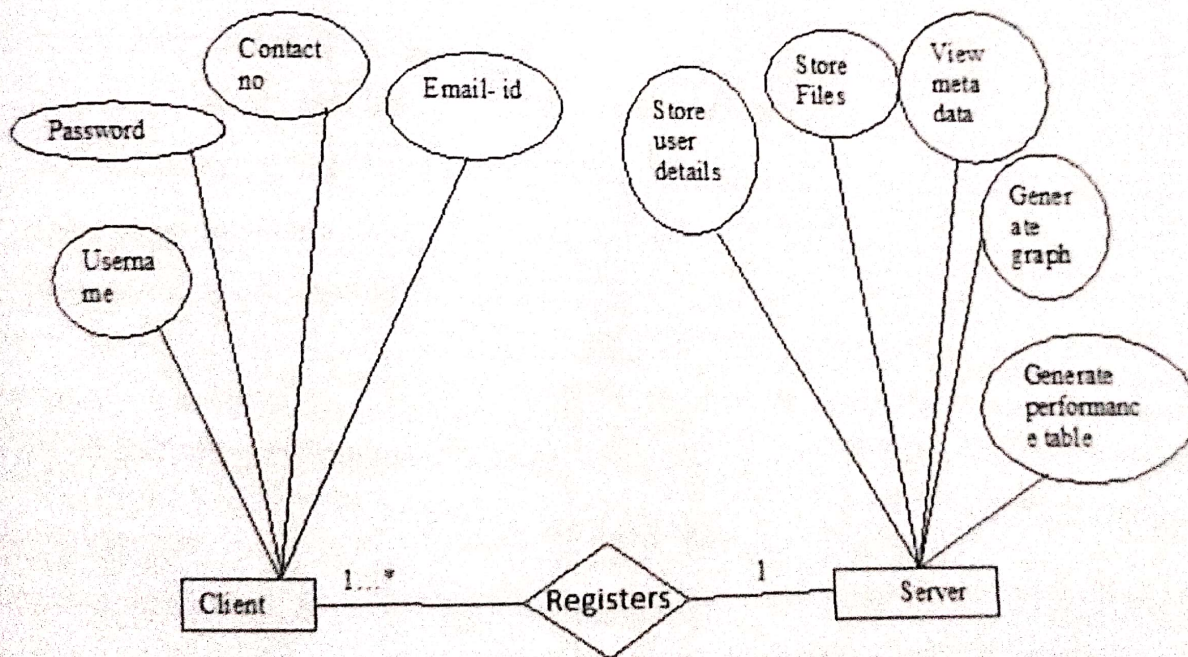
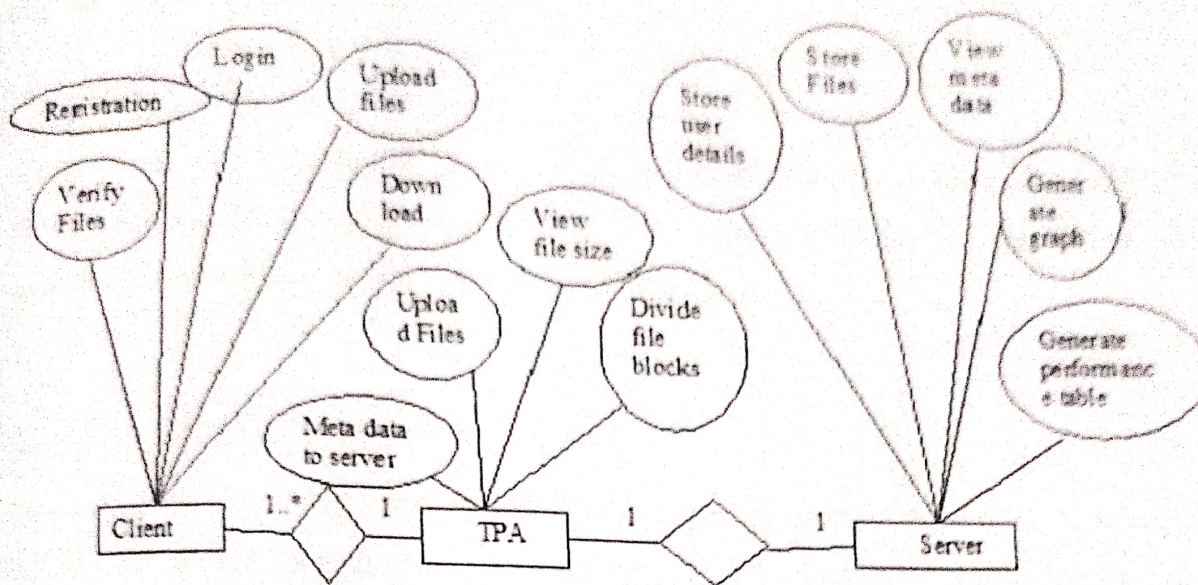


Fig 5.3.6 ER-Diagrams

5.4 SOFTWARE ENVIRONMENT

INTRODUCTION

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has waned with passage of time, while Java's has grown.

5.4.1 APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in software but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like LinkedIn.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

5.4.2 FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

3) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language. On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime environment can run Java Programs.

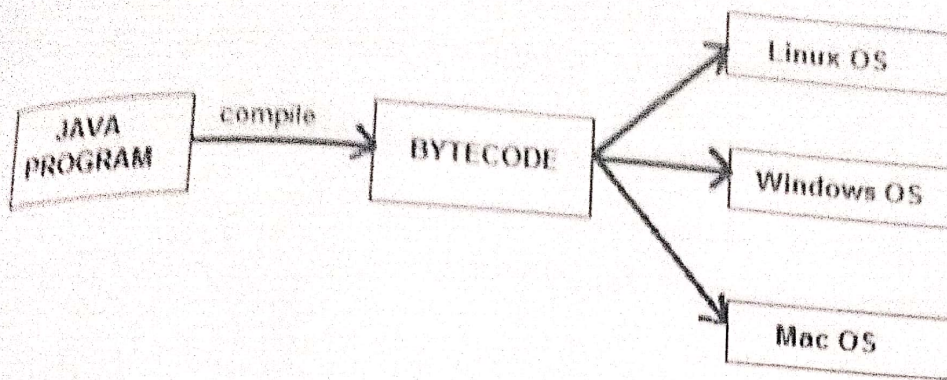


Fig 5.4.2.1 Java Platform Independent

1) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

2) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

3) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

4) Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

5) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

5.1 COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects. Any group of individual objects that are represented as a single unit is known as a Java Collection of Objects. In Java, a separate framework named the "*Collection Framework*" has been defined in JDK 1.2 which holds all the Java Collection Classes and Interface in it. In Java, the Collection interface (`java.util.Collection`) and Map interface (`java.util.Map`) are the two main "root" interfaces of Java collection classes [2].

A framework is a set of classes and interfaces which provide a ready-made architecture. In order to implement a new feature or a class, there is no need to define a framework. However, an optimal object-oriented design always includes a framework with a collection of classes such that all the classes perform the same kind of task.

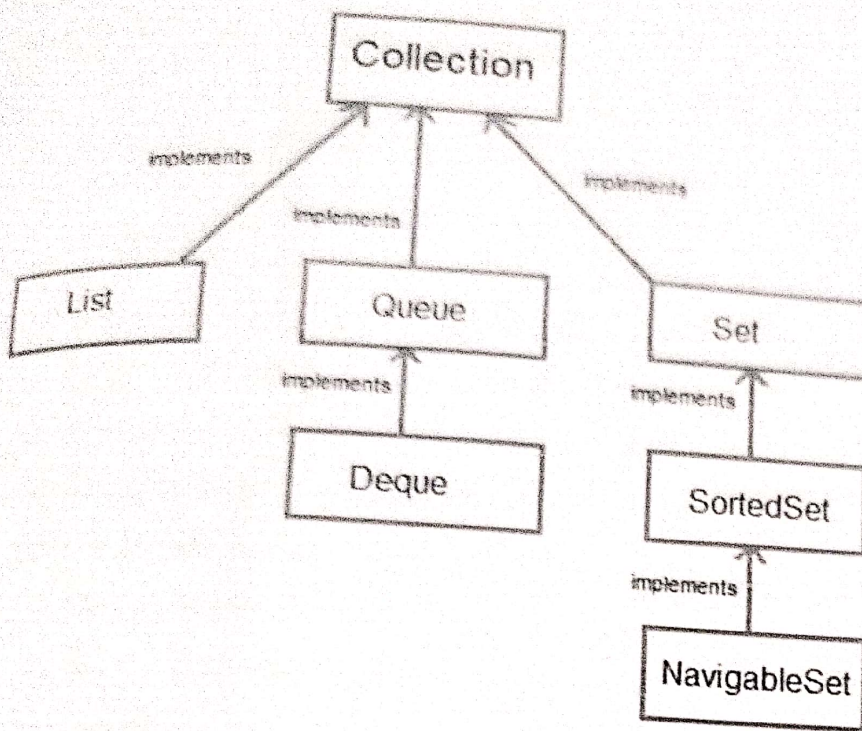


Fig 5.5.1 List of Java Collections

5.6 TESTING PROCESS

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 that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test [13]. Each test type addresses a specific testing requirement. The testing process also include a lot of tests which certainly need a large quantity of network and testing. Based on the availability of testing process the need mustbe significantly maintained and as per the bases the testing is done.

5.6.1 TYPES OF TESTS

- **Unit Testing**

Unit testing involves the design of test cases that validate that the internal Program logic is functioning properly, and that program input produces valid Outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application [14]. It is done after the Completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation and user manuals. Each function is compared to corresponding requirement to ascertain whether its output is consistent with end users expectations should be considered that each application feature works as per the software requirement.

- **Functional testing is centered on the following items:**

Valid Input is used to identified classes of valid input must be accepted.

Invalid Input is used to identified classes of invalid input must be rejected

Functions is used to identified functions must be exercised.

Output is used to identify classes of application outputs.

Systems/Procedures is used to interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive

Processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined [15].

- **System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test [16]. System testing is based on process descriptions and flows, emphasizing pre- Driven process links and integration points.

- **5.6.1.5 White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least it's purpose. It is used to test areas that cannot be reached from a black box level.

- **5.6.1.6 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of

tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirement document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

5.7 TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

Test Objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.7.1 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications.

5.7.2 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements [17]. Acceptance testing is last phase of software testing performed after system testing and before making the system available for actual use.

5.7.3 ALPHA TESTING

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and preloads work.

5.7.4 BETA TESTING

In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out [18]. Beta testing can be considered and quot pre-release testing and quot. Beta test versions of software are now distributed to curriculum specialists and teachers to give the program a & quot real-world & quot test.

CHAPTER 6 RESULTS AND DISCUSSION

6.1 PRESENTATION OF FINDINGS

The deployment of blockchain in cloud computing would typically involve analyzing the outcomes of implementing blockchain technology within cloud computing environments. The deploying blockchain in cloud computing entails a thorough examination of the technical performance, security enhancements, cost efficiency, scalability, and flexibility, alongside regulatory considerations and user experience. Evaluating the technical aspects involves analyzing latency, throughput, and scalability metrics, comparing pre- and post-implementation scenarios. Security enhancements are scrutinized for improvements in data integrity, cryptographic measures, and resilience against attacks. Cost implications are assessed, encompassing setup costs, maintenance expenses, and potential savings or increased costs vis-à-vis conventional cloud setups. Scalability and flexibility are analyzed to understand the impact on resource scaling and adaptability to changing demands. Regulatory and compliance challenges are discussed, focusing on how they affect deployment strategies [19]. User experience feedback and adoption rates provide insights into user acceptance and potential barriers. Challenges encountered during deployment, such as technical complexities or integration issues, are candidly addressed, along with future outlook and recommendations for further research and development, emphasizing areas for improvement and emerging trends in the field. User experience and adoption rates provide valuable insights into the acceptance and usability of blockchain-enabled cloud computing platforms. Feedback from users and stakeholders offers valuable perspectives on user satisfaction, ease of use, and potential barriers to adoption. Understanding user preferences and concerns is vital for refining user interfaces and driving widespread adoption of blockchain technology in cloud environments.

Deploying blockchain technology in cloud computing offers several potential benefits, including enhanced security, transparency, and efficiency in various processes. Below are some possible results and conclusions drawn from such deployments:

6.2 Results

Enhanced Security: Blockchain's cryptographic principles provide a high level of security, making it suitable for securing data and transactions in cloud computing environments. Immutable ledgers ensure that data stored on the blockchain remains tamper-proof, reducing the risk of unauthorized access or data manipulation.

Improved Transparency: Blockchain's decentralized nature facilitates transparency by allowing all participants in a network to access a shared ledger. This transparency can be beneficial in cloud computing scenarios where multiple parties need to access and verify data or transactions.

Increased Efficiency: Blockchain can streamline various processes in cloud computing, such as identity management, supply chain tracking, and smart contract execution. By automating trust mechanisms and reducing intermediaries, blockchain can potentially improve the efficiency of cloud-based operations [20].

Cost Reduction: Through decentralization and automation, blockchain can help reduce the costs associated with intermediary services, data reconciliation, and transaction processing in cloud computing environments.

CHAPTER 7

CONCLUSION

In this project, we propose a cloud auditing system for data storage security in Cloud Computing. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced datafiles, we further extend our privacy preserving public auditing protocol into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient [21].

7.1 RECOMANDATIONS AND FUTURE RESEARCH

7.1.1 Technical Performance: This subsection would discuss the technical aspects of deploying blockchain in cloud computing, such as latency, throughput, and scalability. It would detail the performance metrics before and after the implementation of blockchain and assess any improvements or drawbacks observed.

7.1.2 Security Enhancement: Blockchain is often touted for its enhanced security features due to its decentralized and immutable nature. This section would discuss any improvements in security achieved through the deployment of blockchain in cloud computing environments. It might include discussions on data integrity, cryptographic measures, and resistance to attacks [22].

7.1.3 Cost Efficiency: Blockchain deployment might affect the cost structure of cloud computing operations. This part would delve into the cost implications of integrating blockchain, including any initial setup costs, ongoing maintenance expenses, and

potential savings or increased costs compared to traditional cloud setups.

7.1.4 Scalability and Flexibility: Blockchain deployment might influence the scalability and flexibility of cloud computing systems. This section would discuss how the integration of blockchain impacts the ability to scale resources and adapt to changing demands [23]. It might also explore any limitations or advantages in terms of flexibility compared to conventional cloud setups.

7.1.5 Regulatory and Compliance Considerations: Compliance requirements and regulatory frameworks can significantly impact the deployment of blockchain in cloud computing, especially in highly regulated industries like finance and healthcare. This part would discuss any regulatory challenges faced during deployment and how they were addressed.

7.1.6 User Experience and Adoption: This subsection would evaluate the user experience and adoption rates of blockchain-enabled cloud computing solutions. It might include feedback from users or stakeholders involved in the deployment process and assess any barriers or facilitators to adoption [24].

7.1.7 Challenges and Limitations: No deployment is without its challenges and limitations. This part would candidly discuss any hurdles encountered during the deployment process, such as technical complexities, integration issues, or resistance from stakeholders.

7.1.8 Future Outlook and Recommendations: Finally, this section would provide insights into the future outlook for blockchain deployment in cloud computing and offer recommendations for further improvements or research directions [25]. It might highlight areas for future development, potential optimizations, or emerging trends in the intersection of blockchain and cloud computing.

REFERENCES

- [1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," *Proc. IEEE INFOCOM '10*, Mar. 2010.
- [2] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," <http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html>, June 2009.
- [3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.
- [4] <http://www.cloudsecurityalliance.org>, 2010.
- [5] <http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof-mass-email-deletions/>, 2006.
- [6] Y. Sameena Banu and R. Kumar, "Security in cloud computing using blockchain technology", *3rd National Conference on Image Processing Computing Communication Networking and Data Analytics*, pp. 422, 2018.
- [7] M. Kumar, A. K. Singh and T. S. Kumar, "Secure log storage using blockchain and cloud infrastructure", *2018 9th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-4, 2018.

- [8] M. Samaniego and R. Deters, "Hosting virtual iot resources on edge-hosts with blockchain", *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 116-119, 2016.
- [9] J. H. Park and J. H. Park, "Blockchain security in cloud computing: Use cases challenges and solutions", *Symmetry*, vol. 9, no. 8, pp. 164, 2017.
- [10] S. G. Sharma, L. Ahuja and D. Goyal, "Building secure infrastructure for cloud computing using blockchain", *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1985-1988, 2018.
- [11] S. Pavithra, S. Ramya and S. Prathibha, "A survey on cloud security issues and blockchain", *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 136-140, 2019.
- [12] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*, O'Reilly Media, Inc, 2014.
- [13] J. Barkatullah and T. Hanke, "Goldstrike 1: Cointerra's first-generation cryptocurrency mining processor for bitcoin", *IEEE micro*, vol. 35, no. 2, pp. 68-76, 2015.
- [14] R. Cortés, X. Bonnaire, O. Marin, L. Arantes and P. Sens, "Proceedings 2016 IEEE 15th international symposium on network computing and applications nca 2016", 2016.

- [15] S. Huh, S. Cho and S. Kim, "Managing iot devices using blockchain platform", *2017 19th international conference on advanced communication technology(ICAICT)*, pp. 464-467, 2017.
- [16] A. Sutton and R. Samavi, "Blockchain enabled privacy audit logs", *International Semantic Web Conference*, pp. 645-660, 2017.
- [17] J. Cucurull and J. Puiggali, "Distributed immutabilization of secure logs", *International Workshop on Security and Trust Management*, pp. 122-137, 2016.
- [18] M. Samaniego, U. Jamsrandorj and R. Deters, "Blockchain as a service for iot", *2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber physical and social computing (CPSCom) and IEEE smart data (SmartData)*, pp. 433-436, 2016.
- [19] S. Panikkar, S. Nair, P. Brody and V. Pureswaran, "Adept: An iot practitioner perspective", *Draft Copy for Advance Review IBM*, 2015.
- [20] C. Wressnegger, K. Freeman, F. Yamaguchi and K. Rieck, "Automatically inferring malware signatures for antivirus assisted attacks", *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 587-598, 2017.
- [21] N. Bozic, G. Pujolle and S. Secci, "A tutorial on blockchain and applications to secure network control-planes", *2016 3rd Smart Cloud Networks and Systems (SCNS)*, pp. 1-8, 2016.

- [22] J. S. Cermeño, "Blockchain in financial services: Regulatory landscape and future challenges for its commercial application", *BBVA Research Paper*, vol. 16, pp. 20, 2016.
- [23] S. R. Basnet and S. Shakya, "Bss: Blockchain security over software defined network", *2017 International Conference on Computing Communication and Automation (ICCCA)*, pp. 720-725, 2017.
- [24] H. Cheng, C. Rong, M. Qian and W. Wang, "Accountable privacy preserving mechanism for cloud computing based on identity-based encryption", *IEEE Access*, vol.6, pp. 37869-37882, 2018.
- [25] A. Arora, A. Khanna, A. Rastogi and A. Agarwal, "Cloud security ecosystem for data security and privacy", *2017 7th International Conference on Cloud Computing Data Science and Engineering-Confluence*, pp. 288-292, 2017.
- [26] I. Eyal, A. E. Gencer, E. G. Sirer and R. Van Renesse, "Bitcoinng: A scalable blockchain protocol", *13th USENIX symposium on networked systems design and implementation (NSDI 16)*, pp. 45-59, 2016.
- [27] G. O. Karame, E. Androulaki and S. Capkun, "Double-spending fast payments in bitcoin", *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 906-917, 2012.

APPENDIX

A. SOURCE CODE

```
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;
import com.oreilly.servlet.MultipartRequest;
import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.security.SecureRandom;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.text.SimpleDateFormat;
```



```

import java.util.Date;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.servlet.http.HttpSession;

@MultipartConfig(maxFileSize=1024*1024*50)
public class UploadFile extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        HttpSession session=request.getSession();
        String email=(String)session.getAttribute("email");
        PrintWriter out = response.getWriter();
        final String filepath = "C:\\Users\\user\\OneDrive\\Desktop\\";
        try{
            MultipartRequest m = new MultipartRequest(request, filepath);
            String fid=m.getParameter("fid");
            String subject=m.getParameter("subject");
            File f=m.getFile("file");
            String filename=f.getName();
            long fsize=f.length();

```



```

System.out.println("filename:"+filename)
BufferedReader br=new BufferedReader(new FileReader(f));
    StringBuffer sb = new StringBuffer();
    String temp = null;
    while ((temp = br.readLine()) != null) {
        sb.append(temp)
    }
    String data=sb.toString();
    System.out.println("++++++"+data)
    KeyGenerator keygen=KeyGenerator.getInstance("AES");
    keygen.init(128);
    SecretKey s=keygen.generateKey();
    //converting secret keyto string
    byte[] b=s.getEncoded();//encoding secret key
    String skey=Base64.encode(b)
    String cipher=new encryption().encrypt(data,s);
    Date d=new Date();
    SimpleDateFormat s1=new SimpleDateFormat("dd-MM-yyyy");
    String da=s1.format(d)
    Stringquery="insertinto files
values('"+fid+"','"+email+"','"+filename+"','"+subject+"','"+data+"','"+cipher+"','"+skey+"','"+
da+"','"+fsize+"','Waiting')
    Class.forName("com.mysql.jdbc.Driver");
    Connection
    DriverManager.getConnection("jdbc:mysql://localhost:3306/dynamic_auditing"

```



```

"root", "root");
Statement st = con.createStatement();
if (rs.executeUpdate(query)
    > 0){
    File f1 = new File("C:\\Users\\user\\OneDrive\\Desktop\\", filename);
    FileWriter fw = new FileWriter(f1);
    fw.write(cipher);
    fw.close();
    Ftpcon ftpcon = new Ftpcon();
    ftpcon.upload(f1, filename);
    response.sendRedirect("UploadFile.jsp?msg=success");
} else{
    response.sendRedirect("UploadFile.jsp?msg=failed");
}

response.sendRedirect("Up_Encrypt.jsp")
}

catch(Exception e){
    out.println(e);
}
}

```

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/**

* Handles the HTTP <code>GET</code> method.


```

    * @param request servlet request
    * @param response servlet response
    * @throws ServletException if a servlet-specific error occurs
    * @throws IOException if an I/O error occurs
    */

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */

```



```

*/
@Override
public String getServletInfo() {
    return "Short description";
}

// </editor-fold>
}

<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>

<!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>An Efficient and Secure Dynamic Auditing</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<script type="text/javascript" src="layout/scripts/jquery.min.js"></script>
<script type="text/javascript" src="layout/scripts/jquery.slidepanel.setup.js"></script>
<!-- Homepage Only Scripts -->
<script type="text/javascript" src="layout/scripts/jquery.cycle.min.js"></script>
<script type="text/javascript">
$(function() {
$$('#featured_slide').after('<div id="fsn"><ul id="fs_pagination">').cycle({
    timeout: 5000,
    fx: 'fade',

```


Deployment of blockchain in cloud computing

Sankranthi V S A Santosh
Department of Computer Science
and Engineering
Sathyabama Institute of Science
And Technology
Chennai, India
santosh@gmail.com

Hari Krishna Sai Nidadavolu
Department of Computer Science
and Engineering
Sathyabama Institute of Science
And Technology
Chennai, India
nharikrishnasai@gmail.com

Dr R. Rajalakshmi
Department of Computer Science
And Engineering
Sathyabama Institute of Science
And Technology
Chennai, India
rajalakshmi_cse@sathyabama.ac.in

Abstract. Blockchain technology's potential uses in a variety of industries, including cloud computing, have drawn a lot of attention in recent years. A number of advantages, including better data integrity, enhanced security, more transparency, and decentralized control, come with implementing blockchain in cloud computing. In this study, we analyze the opportunities and problems related to blockchain and cloud computing integration. We go over the many architecture models—private, public, and hybrid—for implementing blockchain in cloud environments. In order to guarantee compatibility with cloud-based apps, we examine the crucial factors to take into account when choosing a suitable blockchain platform, including scalability, consensus algorithm, and smart contract capabilities. We also examine the possible applications of blockchain in cloud computing, including resource allocation, identity management, and data sharing. We also look into the possible downsides and restrictions of using blockchain technology in the cloud, including network latency, processing overhead, and legal issues. In conclusion, we outline a number of future directions and trends that we anticipate will emerge in this rapidly developing field. These include the creation of more efficient consensus algorithms, the incorporation of blockchain technology into cloud computing to integrate artificial intelligence and the Internet of Things, and the investigation of novel blockchain-based security and privacy measures. According to our research, integrating blockchain technology with cloud computing might completely change how we share, store, and handle data while maintaining responsibility, security, and trust. **Keywords:** blockchain, cloud computing, security, transparency, data integrity, decentralized control, architectural models, blockchain platform, scalability, consensus algorithm, smart contracts, use cases, limitations, research directions, future trends.

1 INTRODUCTION

The potential of blockchain technology to transform a number of industries has drawn a lot of attention in recent years. One area where its use offers considerable promise is

cloud computing. The pay-as-you-go internet distribution of computing resources, including software, storage, and processing power, is referred to as cloud computing. By removing the requirement for businesses to operate and maintain their own physical infrastructures, this strategy offers flexibility, scalability, and cost effectiveness.

Blockchain technology usage in cloud computing has a number of benefits. Blockchain technology, first and foremost, offers improved data integrity and security. Blockchain uses cryptographic techniques to make sure cloud data is transparent and impervious to tampering. This is especially important for cloud computing because data security has long been a top priority. Organizations using blockchain technology might feel more secure about the accuracy of their data as any modifications are tracked on a decentralized, irreversible ledger.

Furthermore, blockchain makes cloud computing more transparent and accountable. Because blockchain technology is decentralized, all activities and transactions can be monitored in real time, providing a transparent audit trail. Because it allows for a thorough examination of resource utilization and guarantees that resources are used for their intended purposes, this openness enhances trust between cloud service providers and their clients.

The increased interoperability that comes with using blockchain in cloud computing is another important advantage. Blockchain technology has the potential to serve as a link between various cloud service providers, facilitating the easy exchange and transfer of information between them. Increased efficiency and productivity can result from this since it can facilitate cooperation, cutdown on redundancy, and make it easier to integrate various cloud services.

Furthermore, blockchain makes it possible for cloud computing to have a more effective and economical invoicing and payment system. Conventional cloud billing solutions frequently entail intricate computations and middlemen, which can be expensive and time-

12th INTERNATIONAL CONFERENCE ON CONTEMPORARY ENGINEERING AND TECHNOLOGY 2024

ORGANIZED BY
ORGANIZATION OF SCIENCE & INNOVATIVE ENGINEERING AND TECHNOLOGY (OSIET), CHENNAI, INDIA

IN COLLABORATION WITH
SAMARKAND STATE UNIVERSITY, UZBEKISTAN

IN ASSOCIATION WITH

PRINCE SHRI VENKATESHWARA PADMAVATHY ENGINEERING COLLEGE
PRINCE DR. K. VASUDEVAN COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved By All India Council For Technical Education, Affiliated To Anna University)
Medavakkam - Mambakkam Main Road, Ponmar, Chennai - 600 127.

Certificate of Presentation

This is to certify that Mr/Mrs/Dr. Sankranthi V S A Santosh from
Sathyabama Institute of Science and Technology
..... has presented a paper titled
Deployment of Blockchain in Cloud Computing
..... in the "12th International
Conference on Contemporary Engineering and Technology 2024" held on 23rd & 24th March 2024.


Dr. Akhatov Akmal Rustamovich

Director of International Affairs,
Medavakkam Main Road, Chennai


Dr. Christo Ananth

Professor,
Samarkand State University, Uzbekistan


Dr. P. Aravinthan, M.E., Ph.D.

Technical Director,
Secretary


Prof. A. Krishnamoorthy, M.E., Ph.D.

Technical Director,
Administrator


K. Janani

Coordinator,
Chairman

