TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

CENTRAL CAMPUS

**A**

**PROJECT PROPOSAL**

**ON**

# DECENTRALIZED SECURE CLOUD STORAGE PLATFORM USING BLOCKCHAIN

**Submitted By:**

**Anish Shrestha (071/BCT/504)**

**Bipin Khatiwada (071/BCT/512)**

**Sagar Bhusal (071/BCT/551)**

**Sumit Chhetri (071/BCT/546)**

Submitted To:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
PULCHOWK CAMPUS,LALITPUR
NEPAL

January 13,2018

# ACKNOWLEDGEMENT

We would like to express our notable thanks to **Department of Electronics and Computer Engineering** for providing us environment to dig the latest technology,investigate it and do research on those areas as Major Project.

We would also like to thank **Prof. Dr. Subarna Shakya**, our Project Mentor who is providing us guidance and helping us with different views of application, development and scalability.

Our special appreciation goes to **Er. Sandip Pandey**, Alumni of Pulchowk Campus for guiding us in this project and helping to better cope with the difficulties along with giving supportive hands.Without his valuable guidance and suggestions in Blockchain technology, it would have been difficult for us to come to a conclusion to work in this new technology.

We are also grateful to **Dr. Dibakar Raj Pant**, **HOD**, Department of Electronics and Computer Engineering as well as our **DHODs, Mrs. Bibha Sthapit** and **Mr. Dinesh Baniya Kshatri** for presenting useful suggestions regarding the queries put forward regarding the project.

Last but not least; we would appreciate all our teachers,seniors and friends for their support and constant inspiration which helped us to withstand in our dreams and make it come true.

Anish Shrestha (071/BCT/504)
Bipin Khatiwada (071/BCT/512)
Sagar Bhusal (071/BCT/551)
Sumit Chhetri (071/BCT/546)

# SUMMARY

With the increase of popularity of Cryptocurrencies and dApps in recent time, Blockchain has been growing to be a next big revolutionary technology. No doubt Blockchain will now be implemented in most of the fields in coming years.

The project is to develop a platform where user can store their valuable and sensitive informations more securely in the web such that therell be no need of any organization with whom trust is to be kept. Currently, people used Cloud storage and different storage spaces hosted or controlled by other party. They can have whole authority over users data. Implementing this system, we aim to break this authority of any single organization over users data.

The project deals about development of a platform where user can store credentials such as: passwords, secret keys of their cryptocurrency wallet, sensitive documents, papers etc. It also allows users to either get paid by hosting data or pay for using spaces for data storage through a peer-to-peer network. By using IPFS system as well as Leveldb for decentralized/distributed data access and retrieval, with Kademlia DHT implementation for node mapping and route selection along with Ethereum blockchain to store records of Smart Contracts and the hash value of the Metadata of the uploaded files and the parties involved, we plan to create a Network for Decentralized Data Storage. Use of Ethereum Blockchain provides with a mechanism of assurance and trust for all clients and nodes willing to use application, while security is provided using encryptions based on asymmetric keys.

# Contents

# List of Figures

# List of Tables

# 1 INTRODUCTION

## 1.1 Background

With the growing fields of Information Technology, Internet Of Things and Digitization of every business, organizational work and projects, Information has become the biggest valuable asset for anyone. Data has become the most powerful thing in todays world. With the abundance of data and its ever growing nature, its equally important to store it in an organized way such that its easily accessible and secure. For this purpose Databases are being used as a warehouse to store data.

Database play a crucial role for any individual as well as any organization and business to store its data. Realizing the importance of data and insufficiency of storage, databases are replicated, distributed and backed up in different ways. Individuals store data in the cloud provided by different privately companies. Organizations set up their data centers at different part of the globe to store its data. For the security and bandwidth, data are scattered and replicated to different servers at different places. This seems to provide a good solution for the management of rapidly increasing data. And also ensures data safety. In future, the rate of increment of data is sure to reach high. To cope with it, the current database system needs to be more reliable, safe and available all the time.

## 1.2 Problem Statement

### 1.2.1 Lack Privacy of Data

Different cloud service companies and distributed data centre of organization ensures the data availability and safety. However,most of them have terms that allow the company to edit, modify, access, delete, view and analyze your content. This can be done to provide the best possible service to the client, create advertisement, manipulate it in some way to generate income or use for their own purpose or analysis. Data stored to a private owned database gives several access rights to that company and thus, is not always secure.

### 1.2.2 Data Loss

Storing sensitive data only on local machine or drives can sometime be very lamenting because once they are stolen, lost or destroyed by any other means, user cannot make a recovery. Moreover, most of the personal accounts of Cloud Storage also do not cover the insurance of data, take responsibility in case of data loss due to catastrophic failure as well as ensures data availability all the time. This is well stated on Terms and Conditions of Dropbox, Box, RapidShare,Google Drive, Amazon Cloud Drive, MS Onedrive etc. So, completely relying on data storage on your local machine only or on the cloud storage is just not always safe and genuine.

### 1.2.3 Financial and other losses due to Data hack

Furthermore, storing users sensitive data to cloud is not considered a good option when it comes to the high potential value of that data. Sensitive information here means: user passwords, secret keys of cryptocurrencies wallet, secret and confidential documents, sensitive informations, papers, records of banking transactions etc whose loss or hack can bring a huge disaster for any organization or individual.

## 1.3 Solution Proposed

For the above problem statement, the solution we propose here is to establish a distributed database system which will store data in peer to peer network such that therell be no any central body with right to use and modify clients data.

The data will be shred to multiple chunks, encrypted using different cryptography algorithms, stored at different nodes. No any node will know what data and whose data they are storing. Even if the any attacker hacks into any node and pulls data, itll only receive part of the data which is encrypted. So, its efficiently hard to grab complete data in decrypted form by any attacker. Thus is more secure than Cloud.

For the data storing service that the storage nodes (Postmen) of the network provide, they will be rewarded, and the client will pay for that service. Furthermore, the client and Postman will be bound by the Smart Contracts stored in the Blockchain that will act as a proof and trust layer for data availability and storage. This way the network will sustain by reward and compensation mechanism. The integrity of the data and trust of data storage is managed by using Blockchain.

## 1.4 Objectives

Our primary objective is to develop a system over Ethereum Blockchain which can store the users data in a decentralized database distributed across the peer to peer network. Our objectives can be pointed as :

- To develop a system that can be implemented in real world to store users data in network considering more safety, availability and backup.
- To decentralize the storage mechanism of database and remove the sole right of any private company over user data.
- To contribute the development of a viable, practical product based upon the Blockchain technology. Also, to implement the techniques and benefits of using Blockchain in existing system.

## 1.5 Scope Of the Project

In this project, we propose a limited version of a working decentralised database system using different technologies like IPFS and blockchain. This project however does not include the complete bandwidth optimization algorithms, sophisticated NAT traversal algorithms and complete analysis of the security in each layer of the network stack. The scope of the project is limited to show viability of decentralized database system using blockchain and smart contracts. We'll discuss the optimization techniques in the report at the end of the project but the implementation is out of scope.

Decentralized Secure Storage is a very needed feature in todays world where high value data needs to be securely stored in web.

- The scope of this project can be found in any organisation or individual needs where security of the data or information is paramount. For example: Banks, Individuals, different Organizations etc.
- It can also be used in saving the Bandwidth of the Central Server.
- User can register for the Postman and earn reward coins by renting their Storage Devices. Thus can be also used as a earning source by any node.
- Most of the features developed here can be used fully or partially in many other applications like End to End Messenger, different Storage Service etc.

# 2 LITERATURE REVIEW

## 2.1 Existing Data Storage System

Currently, user uses his/her offline storage devices and other secondary storages for data backup and protection. Most of us often use the cloud service of Amazon, Google, Dropbox, Microsoft and others. A huge amount of users data are stored in cloud which is in fact someones computer or storage devices. Such organization has complete authority over users data. In recent years, trend has increased rapidly in using those data without users acknowledge and permission by those company for their uses and pursue higher benefits from it.

## 2.2 Need of a distributed database

With ever-growing technological advancement and shrinking size with more power devices, there has never been a better time for advancements in an information system made up entirely of distributed devices. Of course we have the internet as an example. However, internet itself maintains a general hierarchy of client-server and a lot of middle men which may or may not be trusted. The devices grow, data grow and so do the need for physical as well as logical means to hold the data. With a new race for powerful organizations to gather as much data as possible for future manipulation and understanding of information in a gigantic scale, there has been an unprecedented search and store of data like never before. Even in personal scale, we associate data to our personal lives as digital data has never been more of our personal lifes metadata like today. Hence, they say data is the new currency, data is the new knowledge. [1] [2]

However, how do we store our personal data? We store it locally, we burn it into DVDs, we use cloud services, we store it as much as we can and then leave the rest for trusted servers to put. We have come to an age where our crucial data - we store it in some servers storage space provided by a company with a promise security and integrity. People now pour more trust in such services than themselves to store our data with security and integrity. However, in every system with central authorities there is a hierarchy of power and in this case such misuse directly affects our crucial data we use in our lives. Giving the key to an entity of higher authority in a system can not last long without an imminent risk. One can not guarantee the replication and misuse of such data by such authorities.

However, with the technological advancements, computation power, data transfer rates and storage space distributed to each and every people in forms of many devices. there need not be networks with central authorities and hierarchies to where we trust our data for safe keeping and transfer. Instead we can do what we have been doing for the past decade now in completely distributed systems maintained by parties involved using the system and with Blockchain technology we can do this with trust.

## 2.3 Necessity of blockchain for a distributed storage system

In 2008, after the global financial crisis, a new paper brought about by person or a group named Satoshi Nakamoto introduced a concept of a peer-to-peer distributed currency with a paper titled Bitcoin: A peer-to-peer Electronic Cash System [2]. With there usually being a central authority bearing a higher power in a system may it be in economics, finance, technological or general, there exists a risk of the central authority misusing its power for ones gain while others loss. The paper talked about a distributed digital crypto-currency which could improve how we did things and many regarded as a better system.

Hence, Blockchain with its social perceptions related to Bitcoin not only solves the problem of Double Spending but provides us a solution for a problem with this specific criterias:

- Possibility of Fraud.
- Intermediaries or a middle man.
- Throughputs (Number of transactions/sec).
- Stable data.

Hence, called the FITS model which defines a environment where there is a possibility of Fraud, a middle man for transactions and exchange of other resources, transactions occurring in many number per seconds and a stable data i.e. data that is not constantly changing.

However, taking this step further, Blockchain allows for what is known as a Distributed Autonomous Organization where it is an organization self-sustained by members without a central authority and trust built around what is known as Smart contracts. [3]

Hence, using blockchain technology we can build a system far exceeding the typical client server system in terms of application and scope but with the same amount of trust we put in a centralized server. Like a cloud platform where you are storing your data. You will trust the server of the given cloud service provider to hold the data without tampering and betrayal of trust and make it available for use any time you want. This trust is built by the company itself with continuous service and a good policy. However, in a distributed service this is hard to do as there is no central authority

However, with blockchain we can implement a distributed system with a trust system. The same kind of trust people put in data storage services like a dedicated server or even simple cloud storage applications like Google Drive. Hence, for an implementation of a distributed storage network maintained by the people and for the people to use without an authority that provides trust but have trust built into the system, technology like blockchain is crucial. [4]

## 2.4 What does the Blockchain store?

In this case using the IPFS system we can generate hashes of file that give us access to files in the network. In blockchain, we store the hashes of files - metadata and its hash, the transactions between users, who the data belongs to, who is storing the data, which other parties are involved during the replication and storage process and access control data.

## 2.5 How security is provided against data tampering?

Encryptions with Asymmetric keys paired with Diffie Hellman key exchange and shared secrets, we will implement a layer of security and restriction. Using such approach, one can be sure that ones data is safe and not accessible and readable by undesirable parties.

## 2.6 Existing Decentralized Database System using Blockchain

Although blockchain technology has been developed more than 2 decades ago, its actual implementation in various technology fields is just blooming. There are few such examples of distributed database using blockchain. Some of the most popular applications of blockchain in distributed database management are storj.io, bigchaindb etc. We here discuss BigchainDB,Storj.io , Sia and MaidSafe as it is more somewhat related to our project.

BigchainDB: a scalable blockchain database that uses Blockchain technology to store the users data in various nodes.. BigchainDB inherits characteristics of modern distributed databases: linear scaling in throughput and capacity with the number of nodes, a full-featured NoSQL query language, ecient querying, and permissioning. Being built on an existing distributed DB, it also inherits enterprisehardened code for most of its codebase. [5]

Storj aims to become a cloud storage platform that cant be censored or monitored, or have downtime. It is the first decentralized, end-to-end encrypted cloud storage that uses blockchain technology and cryptography to secure users files. Storj is a platform, cryptocurrency, and suite of decentralized applications that allows you to store data in a secure and decentralized manner. Files are encrypted, shredded into little pieces called 'shards', and stored in a decentralized network of computers around the globe. [6]

Filecoin and Sia are other two decentralized database systems that both support smart contracts on the blockchain that set the rules and requirements for storage, while Storj does not; Storj users pay what they use. This particular payment model means that if a user disappears, the host will no longer be paid for lending their space, a potential problem for those who will be renting their storage space. [7]

MaidSafe also aims to do more on its network than trade storage; it markets itself as a crowdsourced internet, on which not only data is stored but decentralized applications live. Miners rent out their unused computing resources to the SAFE network, including hard drive space, processing power and data connectionŁŁand are paid in the native Safecoin. The SAFE network also supports a marketplace in which Safecoin is used to access, with part of the payment going to the applications developer. Miners can also sell the coins that they earn for other digital currencies, and these transactions can happen either on the network or directly between individuals. [7]

Storj, Sia, MaidSafe and Filecoin are all built with a native storage marketplace where users and hosts can buy and sell storage space. All use mining to provide computing power for the network. In Filecoin, not only are miners given token rewards for hosting files, but they must prove that they are continuously replicating the files for more secure storage. They are also rewarded for distributing content quicklyŁŁthe miner that can do this the fastest ends up with the tokens. In Maidsafes networkŁŁdubbed SAFEŁŁSafecoin are paid to the user as data is retrieved; however, this is done in a lottery system where a miner is rewarded at random. The amount of Safecoin someone can earn is directly linked to the resources they provide and how often their computer is turned on. [7]

## 2.7 How our Project will differ from Existing systems?

Many of the system workflow matches with the aforementioned system protocols. However, we plan to build a even more superior system that is far better than the existing systems. Our system will be differ from them in following manners:

- It is more focused on storage of keys, passwords, secret keys of cryptocurrency wallets, credentials , important and secure file systems. So, our system will focus more on securing users data than just providing mass data storage. However, it can be used to store any amount of data.
- Since we are developing it over Ethereum blockchain, Ethereum coins can be used to pay and receive in transaction. Since it is widely used currency. Users do not have to invest in ICO separately to buy coins specially for this system.
- Since the transaction period of Ethereum is less (20 seconds in ideal case) , iit will be relatively faster than other systems like Storj.

# 3 METHODOLOGY

## 3.1 System Architecture

### 3.1.1 User Level Architecture

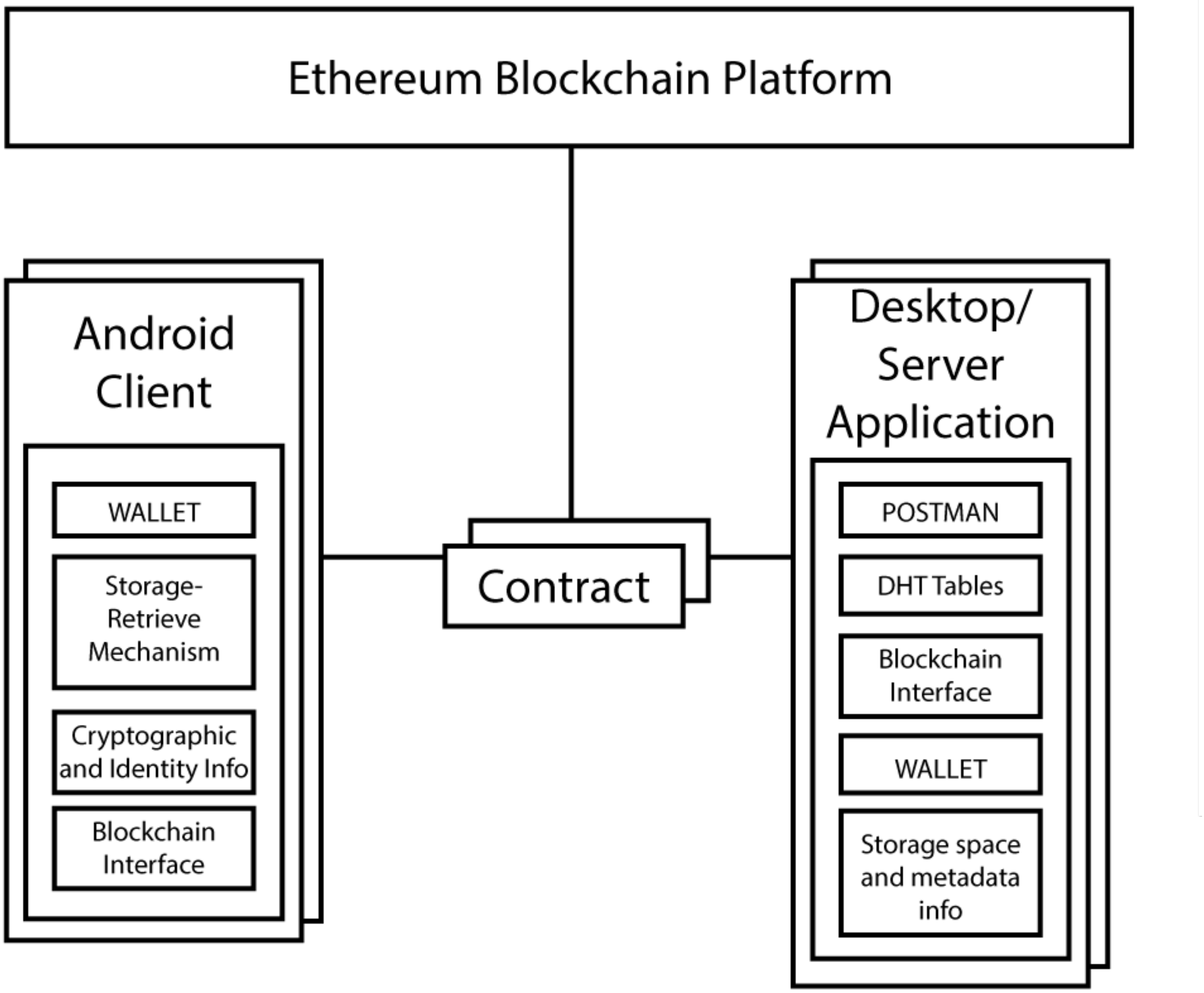

Ethereum Blockchain Platform
Android Client
WALLET
Storage-Retrieve Mechanism
Cryptographic and Identity Info
Blockchain Interface
Contract
Desktop/ Server Application
POSTMAN
DHT Tables
Blockchain Interface
WALLET
Storage space and metadata info

Figure 1: User Level Architecture

### 3.1.2 Core Architecture

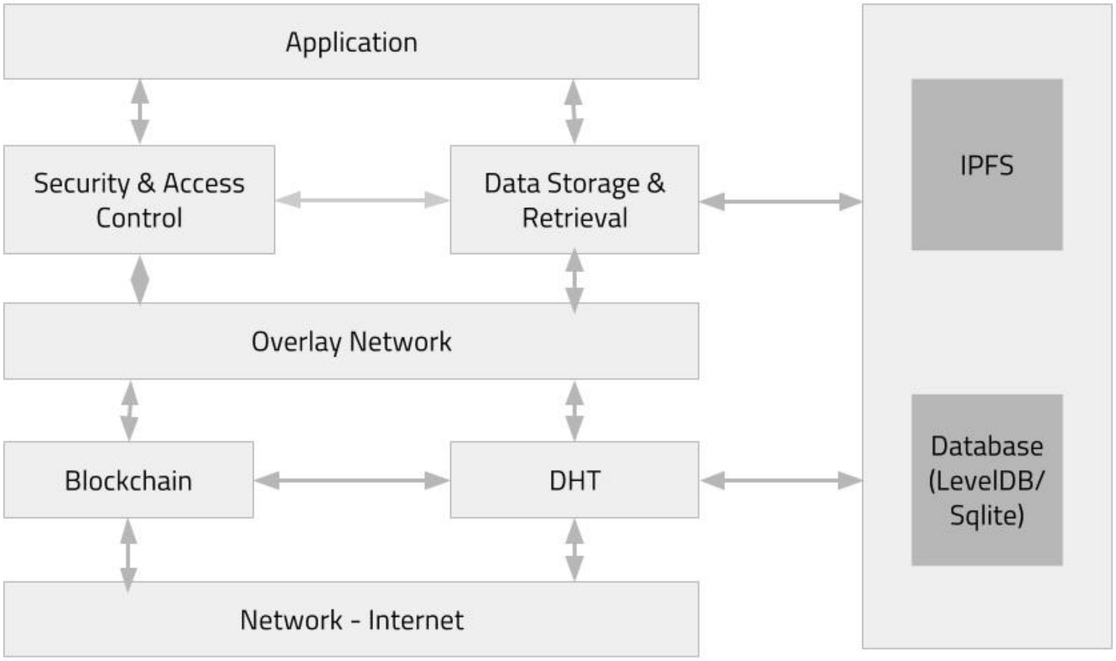


Figure 2: System Architecture

The figure above shows our basic block diagram of the System Architecture. The details of each part is described below.

### 3.1.3 Peer to Peer Network

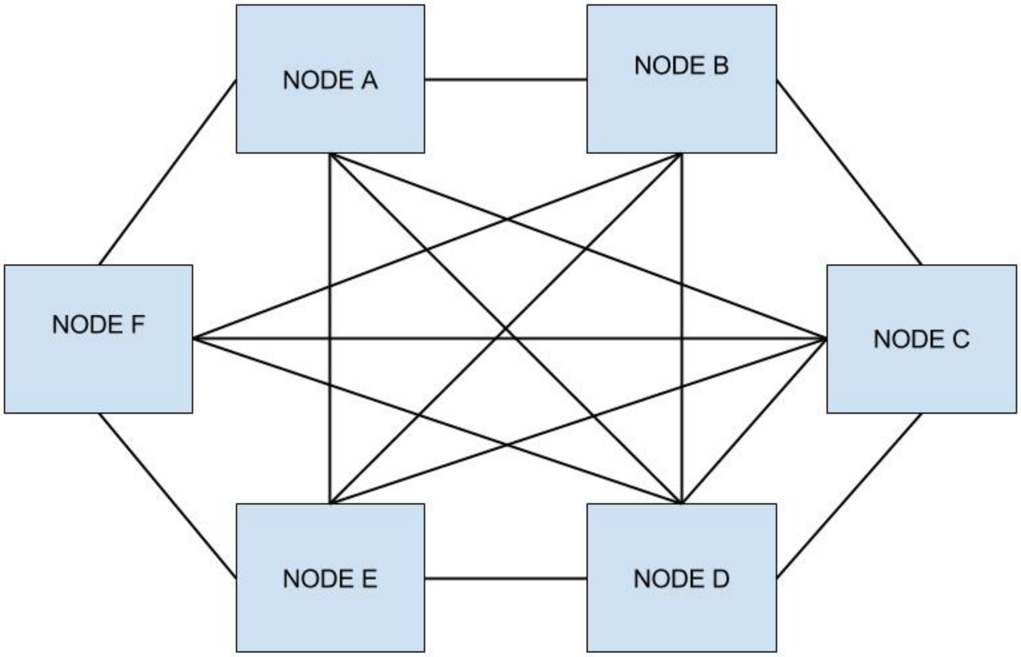


Figure 3: P2P Network

Peer to Peer network is the distributed network where each node in the network communicates with each other directly or through a series of channels via other nodes. Theres no client server to access the resource. Each node will act both as a host or a client as needed. There is no any Central server for controlling the system flow and other nodes.

### 3.1.4 Client, Postman, Sub Postman

Any node or user that uses the database to upload or download data is Client. Postman is the node of the network which is responsible for the storage of data of the client. Postmen which are used to store the copies of data on behalf of some other Postman is SubPostman.

### 3.1.5 DHT and Kademlia

A distributed hash table (DHT) is a dictionary service or more specifically a Mapping Service that stores the different nodes addresses based upon the certain protocol. It provides access to a common shared key-value pair data, distributed over participating nodes with great performance and scalability. On general, it provides two basic functionalities : put(key, value) and get(key) for storing and retrieving node addresses respectively.

There are various implementation of DHT such as Cord, Pastry, Kademlia etc. Due to efficient searching algorithm and less data structure of Kademlia,it is used as DHT.

### 3.1.6 IPFS

InterPlanetary File System (IPFS) is a protocol designed to create a permanent and decentralized method of storing and sharing files. It is a peer to peer network where any node connected to this network can upload file and get all the hash values of the chunks of files. User can note just the generic hash value (root hash). This hash can be used as a tracking id to access the file from any other nodes. Several nodes in the IPFS network form a distributed file system. It has benefit that multiple copies of same file will generate the same hash so that they all can be accessed with the same single hash. Thus, theres no any redundancy of data.

### 3.1.7 LevelDB and Sqlite

LevelDB is an exciting new entrant into the pantheon of embedded databases, notable both for its pedigree, being authored by the makers of the now mythical Google MapReduce and BigTable products, and for its emphasis on efficient disk based random access using log-structured-merge

(LSM) trees.They are mostly useful to optimize random I/Os at insertion/delete time at the price of a slightly degradation of read access time. They are extremely efficient at indexing data in random order stored on rotational disks (i.e. better than b-trees). Data is written in sorted order. LevelDB has been Open Sourced. [8]

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform - we can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures. These features make SQLite a popular choice as an Application File Format. [9]

### 3.1.8 Blockchain

Blockchain is a ever growing chain of blocks that contains immutable records which are linked and secured via Cryptography principles. Each block contains pointer to the next block along with timestamp and transactional data. Blockchains are resistant to modification of the data. It can be implemented in different fields where data is needed to be (or could be) stored permanently and made available to every users of the network. Thus, Blockchain implemented in any system can keep trust, transparency of transactions as well as decentralizes the systems tasks.

### 3.1.9 Ethereum Blockchain

Ethereum is an open-source, public, blockchain-based distributed computing platform featuring smart contract (scripting) functionality. The value token of the Ethereum blockchain is called ether. We will develop our system over the Ethereum blockchain.

### 3.1.10 Solidity

Solidity is known as a contract-based, high-level programming language. This platform has similar syntax to the scripting language of JavaScript. Solidity as a programming language is made to enhance the Ethereum Virtual Machine. Solidity is statically typed scripting language which does the process of verifying and enforcing the constraints at compile-time as opposed to run-time.This typed programming languages will help and do the checking at run-time as opposed to Compile-time. This platform also supports inheritance in object-oriented programming, inheritance enables new objects to take on the properties of existing objects [10]

### 3.1.11 Smart Contract

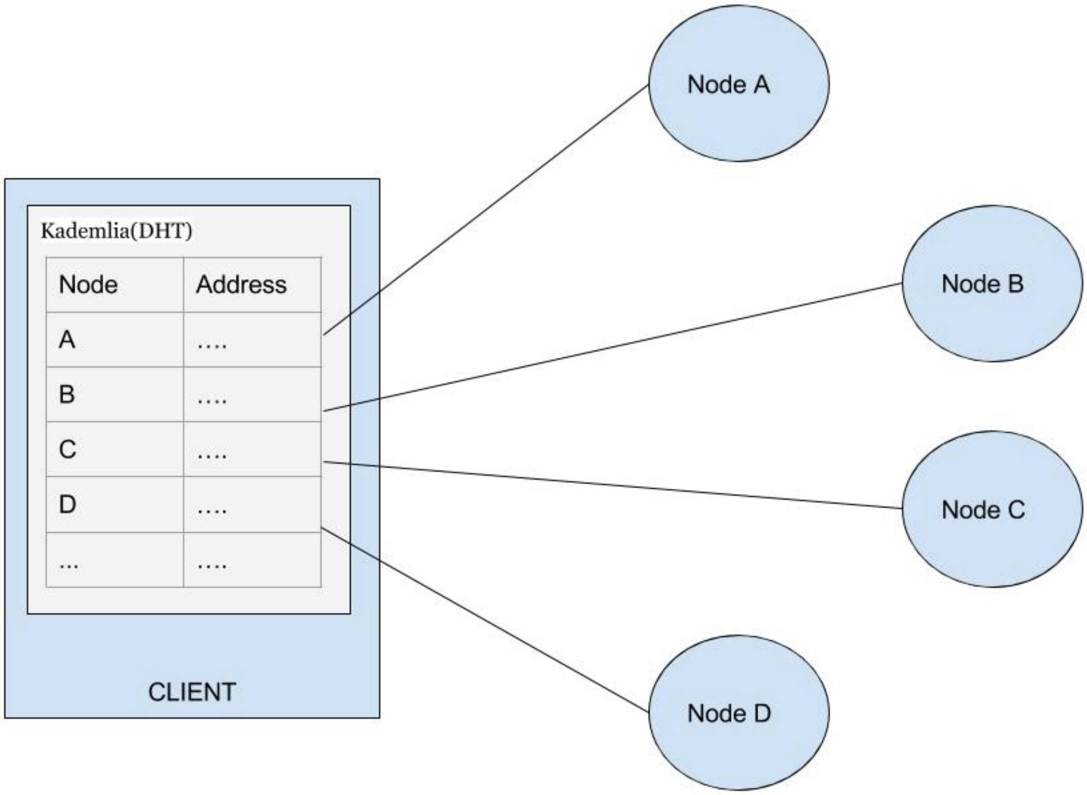
A smart contract is basically a little program that uses computer code to execute a transaction, based on the conditions within the contract. These smart contracts are stored and replicated and ran on a blockchain (a distributed ledger), resulting in ledger updates (Ethereum transactions). When a smart contract has been deployed to the Ethereum network, anyone can call the functions of the smart contract. The function may have security features that block people from using it, but you are free to try. Calling a function on a smart contract is in many ways similar to normal programming - with some differences in executing of course.

### 3.1.12 Rinkeby and Mist Browser

Rinkeby is a test network for deploying ethereum based smart contracts. As ethers hold real world value, this testnet is used as a test ether. Mist is a powerful Ethereum special-purpose browser. It offers like a overall view of the Ethereum blockchain and all needed tools to interact with the blockchain component like Ether, DAO, smart contracts. It is the indispensable tool for running or managing blockchain-specific Apps for the average user who doesnt need to understand technical aspects neither run Command Line Interface.

## 3.2 System Operation/Algorithm

### 3.2.1 Creating Network and DHT



| Node | Address |
|---|---|
| A | .... |
| B | .... |
| C | .... |
| D | .... |
| ... | .... |

Figure 4: DHT

Initially user registers to the network. Its address will be stored in the DHT. A Peer to Peer network will be formed combining all such nodes connected to the network. Each client will get his private and public key. The address can be generated from its public key or public key itself can be used as its address. The system will use Kademlia to store and find out the node address and thus create a channel for communication. The DHT will be used to identify the nodes required and store or retrieve the data. Thus, the network will basically be made of 2 parties: those who host space and those who use provided space for data storage. Operations of data flow and storage between such parties is bound by Smart Contracts. Such transactions are done with the exchange of tokens which is a virtual currency of the network.

### 3.2.2 Storing and Retrieval Of Data (Overlay Network)

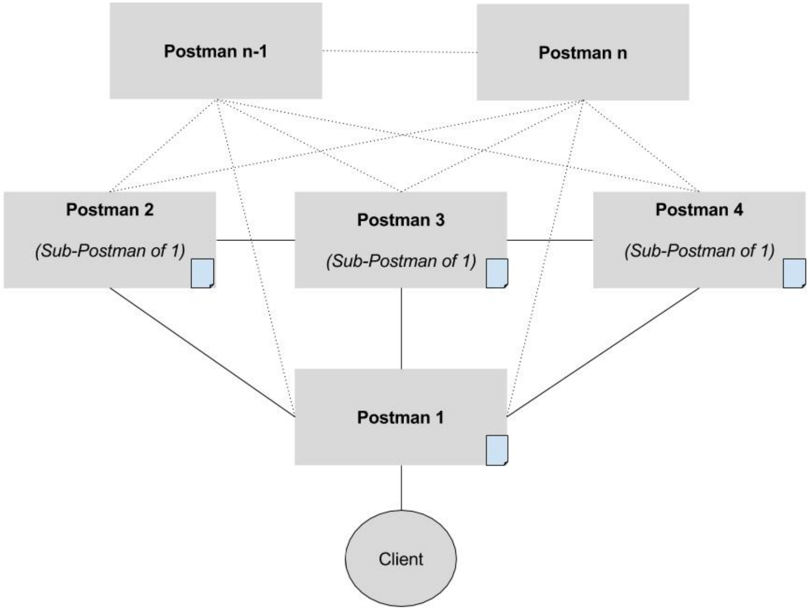


Figure 5: OverlayNetwork

The data that a client uploads will be held by the Postman after they sign Smart Contract which will include terms and conditions that the both party must agree upon. The contract will cover the time duration for data storage, volume that the postman will offer and payment history or deal. The Postman will again have Smart Contract with other Postmans regarding the data availability of client in the absence of it. The data of the client is then replicated to those Sub-Postmen. This ensures that the data is available even in the absence of the main postman to which the client is subscribed.Ethereum Blockchain will be used to store all the transaction history.During retrieval, client can request the data to its Postman and get it. In the absence of its Postman, other Sub Postmen which whom the clients subscribed Postman has formed the smart contracts will deliver the data to that client.

### 3.2.3 Earn by Storing Data

Any user can be connected to the network as a Postman. S/he can store other users data and earn a significant amount for his/her service. The amount of earning will depend upon volume the user stores and the bandwidth it has provided. This will be governed by Smart Contracts. The complete development of this part is not guaranteed by the deadline of this project. However, initial phase will be completed.

### 3.2.4 Security

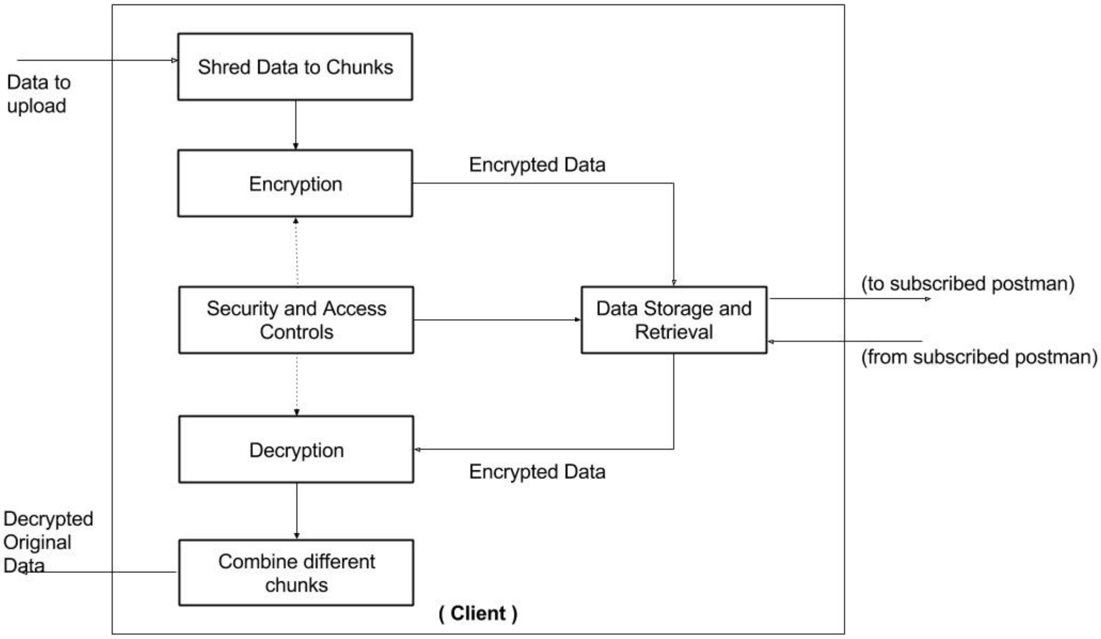


Figure 6: Security

The data that the client uploads will be first divided into chunks and then encrypted by using certain Cryptography Protocols thus ensuring the security and privacy of data. This encryption and decryption will be handled by the Application Interface. As access and permissions to files in the system is restricted using asymmetric encryption and digital signatures along with any other form of identities. There can be data which can only be accessed with specific keys from specific nodes and datas which can be accessed with combination of keys from different nodes. With the help of various encryption algorithm the data are stored in non-readable form so that unauthorized one in network doesnt knows what data is being added and retrieved. For retrieving the data, all the different parts are decrypted and then combined. Security is provided in two different ways:

- The data is divided into chunks which are encrypted in different nodes. So, even if the hacker hacks to a node and access user data, it will be insufficient to generate the complete data from that part without the knowledge of other nodes where remaining parts are stored.
- Each portion of data is encrypted using some cryptography protocol, so intruder cannot

access to the exact data without the decryption. Only the original and valid user can decrypt it, since only she/he knows the passphrase or security key. This will be handled in Smart Contract.

### 3.2.5 Distributed and Decentralized Network

Distributed networking is a distributed computing network system, said to be distributed when the computer programming and the data to be worked on are spread out across more than one computer. Usually, this is implemented over a computer network. Decentralized network usually defines a peer-to-peer architecture where a single point of control, authority and access is omitted.

### 3.2.6 IAAS

Infrastructure as a service (IaaS) is a cloud computing offering in which a vendor provides users access to computing resources such as servers, storage and networking. Organizations use their own platforms and applications within a service providers infrastructure.Following are the key features:

- Instead of purchasing hardware outright, users pay for IaaS on demand.
- Infrastructure is scalable depending on processing and storage needs.
- Saves enterprises the costs of buying and maintaining their own hardware.
- Because data is on the cloud, there can be no single point of failure.
- Enables the virtualization of administrative tasks, freeing up time for other work.

### 3.2.7 Cloud Storage

Cloud storage is a model of data storage in which the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running.

People and organizations buy or lease storage capacity from the providers to store user, organization, or application data.However, with the emergence of distributed cloud storage, the dependency to a centralized authority is ever decreasing.

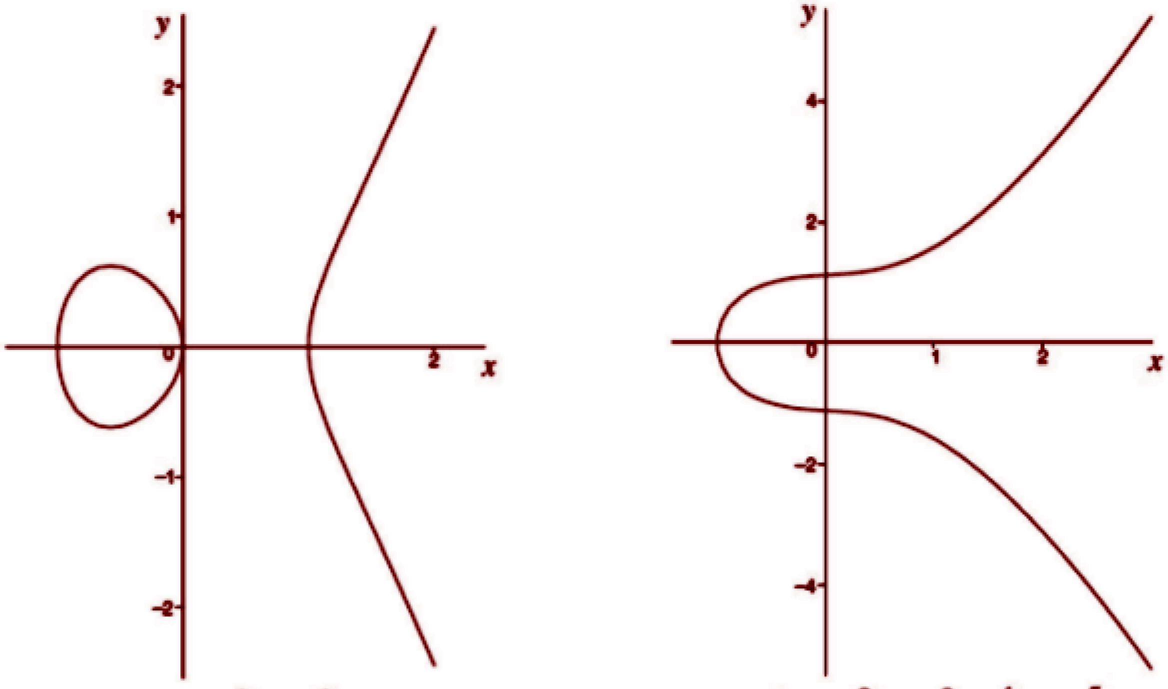
## 3.3 Cryptography

It is a method of changing the plain message or sentences to a scrabble sequence of characters such that it do not possess any sense without decrypting back to original form. This method is applied to convert plain message into secure form so that only the intended party can decrypt it, read and process on it. Cryptography is the major backbone of today's information security. It mainly focuses on four objectives : Confidentiality, Integrity, Non-repudiation and Authentication. Several algorithms exists to perform the cryptography. Their security level, computation power and efficiency also varies. Cryptography can be categorized broadly into three parts : Secret Key Cryptography, Public Key Cryptography and Hashing. We will be dealing with all three parts in our project one way or another.

### 3.3.1 Public and Private Key Pairs

Pubic and Private Key pair are two uniquely related cryptographic keys. Neither public key nor private key will ever have multiple pairs. Private keys are often called Secret or Secret Key. They are kept only by the user and is not shared to any one. Private key can generate signature of a message and only its corresponding public key can verify it. Public keys are often called Sharing Key or Address as it can be made public to other users. Any message encrypted using public key can only be decrypted by its private key. That means to securely send the message to a user, we can lock it using his public key. This locked message can only be unlocked by that particular user since it can be only unlocked using his private key. This is the main feature because of which this has been more popular nowadays. Elliptic Curve algorithm is one of the most used algorithm to generate such key pairs.

### 3.3.2 Elliptic Curve (EC) Cryptographic Algorithm



(a) $E_1 : y^2 = x^3 - x$

(b) $E_2 : y^2 = x^3 + \frac{1}{4}x + \frac{5}{4}$

Figure 7: Elliptic Curve

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. ECC is based on properties of a particular type of equation created from the mathematical group (a set of values for which operations can be performed on any two members of the group to produce a third member) derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations based on elliptic curves have a characteristic that is very valuable for cryptographic purposes:

they are relatively easy to perform, and extremely difficult to reverse.

### 3.3.3 Diffie Hellman Secret Sharing

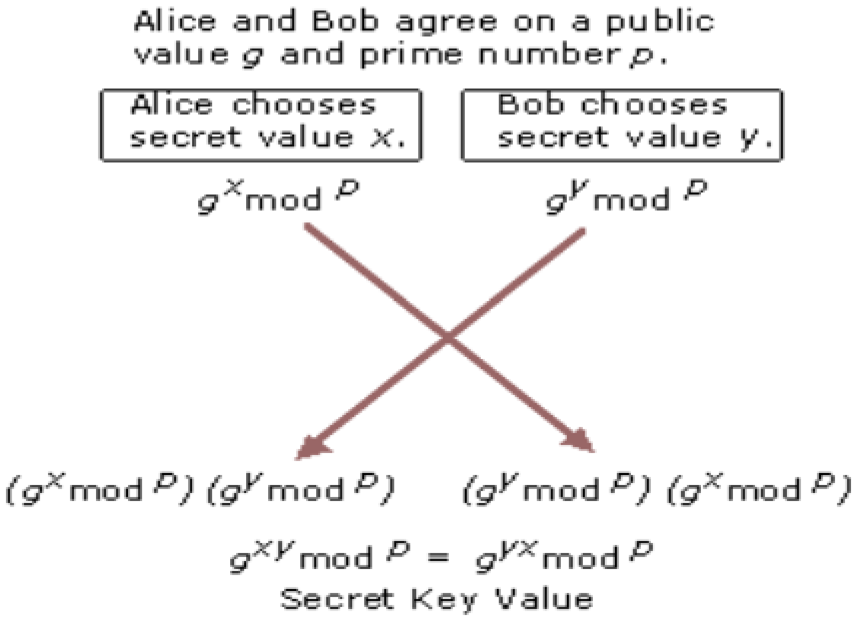

Alice and Bob agree on a public value $g$ and prime number $p$.
Alice chooses secret value $x$.
Bob chooses secret value $y$.
$g^x \bmod p$
$g^y \bmod p$
$(g^x \bmod p)(g^y \bmod p)$
$(g^y \bmod p)(g^x \bmod p)$
$g^{xy} \bmod p = g^{yx} \bmod p$
Secret Key Value

Figure 8: Diffie Hellman Secret Sharing

Diffie-Hellman is a way of generating a shared secret between two people in such a way that the secret can't be seen by observing the communication. That's an important distinction: You're not sharing information during the key exchange, you're creating a key together.

This is particularly useful because you can use this technique to create an encryption key with someone, and then start encrypting your traffic with that key. And even if the traffic is recorded and later analyzed, there's absolutely no way to figure out what the key was, even though the exchanges that created it may have been visible. This is where perfect forward secrecy comes from. Nobody analyzing the traffic at a later date can break in because the key was never saved, never transmitted, and never made visible anywhere.

### 3.3.4 Hashing

Hash is a unique set of characters or an array of bytes derived from a function which intakes certain message or plain text. Each message has unique hash that depends on the hash function used to derive the hash. A slight change in any character of input will produce an entire different hash, meaning that it is impossible to trace any pattern in hashing and also impossible to obtain the original text from it hash value. The hash value is always of same size in most of the cases. For our project, we will be using SHA-256 hashing.

The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function it cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available.

### 3.3.5 Signature Generation and Verification

Digital signature is a unique value that the sign function generates for the given input sequence. Signature of any message can be generated using users private key. The verification can be done by using the message, signature and users public key. Therefore it can make sure that the message has been sent by the user having the private key only. Several libraries available for signature generation and verification.

# 4 REQUIREMENTS

## 4.1 Software requirements

- Integrated Development Environment (IDE) for java and application development.
- Cryptography libraries for Java.
- Development Environment for Kademlia.
- MetaMask - a browser extension for accessing Ethereum enabled distributed applications, or ”Dapps”.
- IFPS Network implementation.
- Ethereum Wallet and Mist Browser.
- Development environment for Solidity - a language for smart contracts implementations in Ethereum blockchain.

## 4.2 Resource requirements

- High bandwidth internet service.
- Reports of past research on distributed networks, DHT and Network Routing.
- Local LANs for testing.

# 5 IMPLEMENTATION PLAN

## 5.1 Infrastructure

Our system will sustain without any dedicated servers when there are enough nodes. However, in the beginning of the system deploy, we might have to set few nodes to facilitate the service. But as the network grow, we do not need any dedicated nodes for data store.

## 5.2 Target Audience

Out target audience is narrowed down to two categories. First one, any normal user who needs to store his sensitive credentials more securely without trusting any cloud storage company. Credentials may include : User passwords, Secret keys of Cryptocurrency wallets, certificates, ownership papers, important documents etc. Second category of target audience is the organization who wants to store their valuable and important assets more securely in the network. For example, IOE might want to use this system to store its question bank pool, past student records etc.

## 5.3 Sustaining the network

Network will sustain because it will give reward for the nodes storing data. Nodes will get paid for the data hosting service. This way, user can join the network, rent his space and earn on the basis of volume it has stored and the bandwidth of data it has provided.

## 5.4 Why choose our system over cloud ?

Since the system is decentralized, no any particular organization will hold the right over users data. So, misuse and use without users permission is removed. Thus, making data more secure. Crash of any node will not cause the data loss since data are replicated and stored at various nodes. No fraud by any node, since the transaction history is recorded in blockchain. Secure in the sense that files are encrypted and shred into chunks, which are then stored at various locations. So, hacking of one node wont reveal users information in any way.

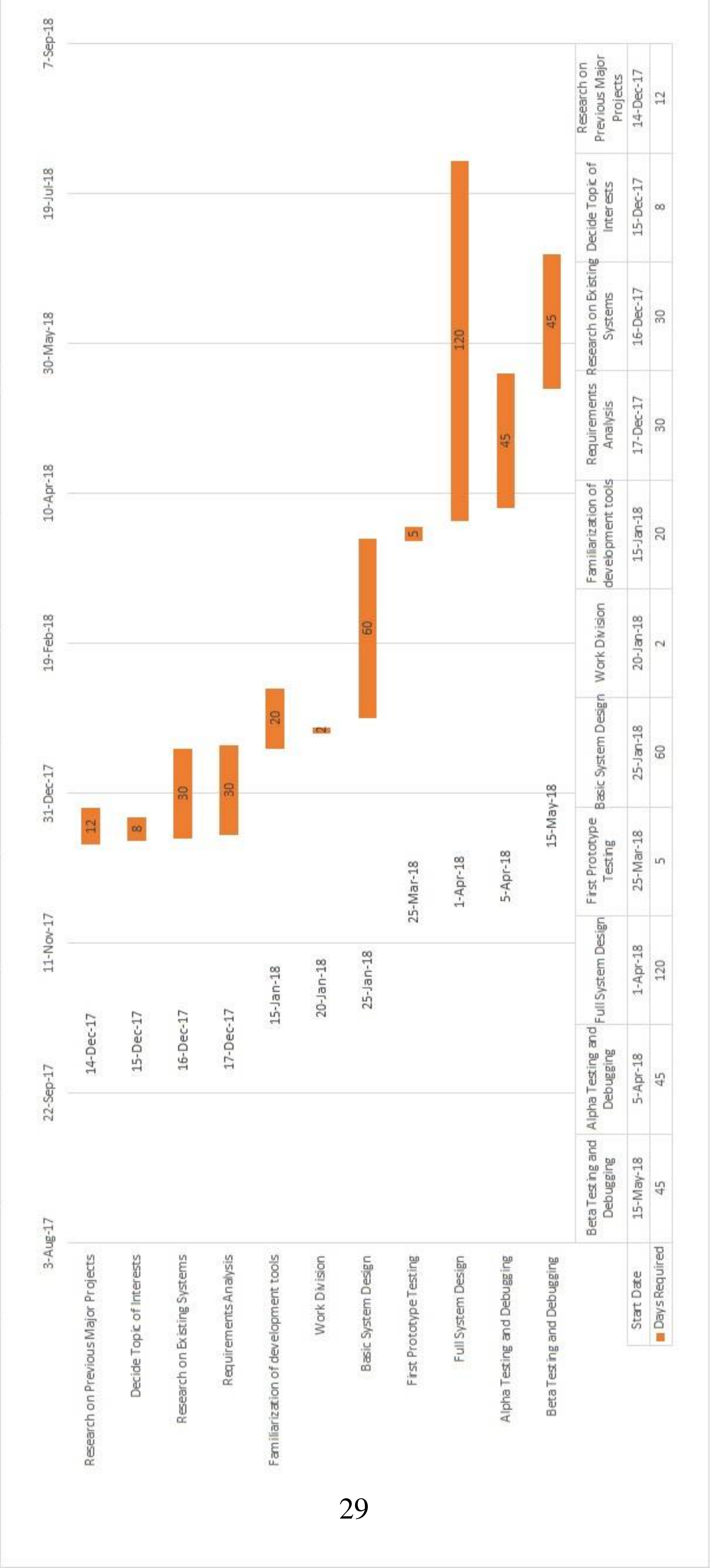
## 5.5 Deliverables

By the completion of this project, we will develop an Android App. Our app will provide an interface to the user where he or she can register to the network. User will get his secret key which will be needed later if he/she wants to login to the system.User can then upload his/her credentials to the network. As a pay for that service, he or she will also have to store some data of other users or pay for the service. The data flow across the network, cryptography, data sharing will all be handled by the inner layer, so user need not have to worry about that. App will also allow user to view his data stored in network and allow him to pull that from network.

# 6 EXPECTED OUTCOME

After the completion of this project, we expect to present an android applicaton to upload and receive files from the decentralized network.We also expect to present server side application for establishment of nodes in the network through which we can host the storage space.The details about data storage and security will be written in contract between the client(android app) and newtork node which will be stored in Ethereum blockchain.

# 7 PROJECT SCHEDULE



| | Beta Testing and Debugging | Alpha Testing and Debugging | Full System Design | First Prototype Testing | Basic System Design | Work Division | Familiarization of development tools | Requirements Analysis | Research on Existing Systems | Decide Topic of Interests | Research on Previous Major Projects |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Start Date | 15-May-18 | 5-Apr-18 | 1-Apr-18 | 25-Mar-18 | 25-Jan-18 | 20-Jan-18 | 15-Jan-18 | 17-Dec-17 | 16-Dec-17 | 15-Dec-17 | 14-Dec-17 |
| ■ Days Required | 45 | 45 | 120 | 5 | 60 | 2 | 20 | 30 | 30 | 8 | 12 |

Figure 9: Project Schedule

# 8 EXPECTED BUDGET

| Title | Expected Budget |
|---|---|
| Meeting Cost | $100.00 |
| Ethereum Cost | $200.00 |
| Travel | $50.00 |
| Stationary | $70.00 |
| Miscellaneus | $150.00 |

Table 1: Expected Budget

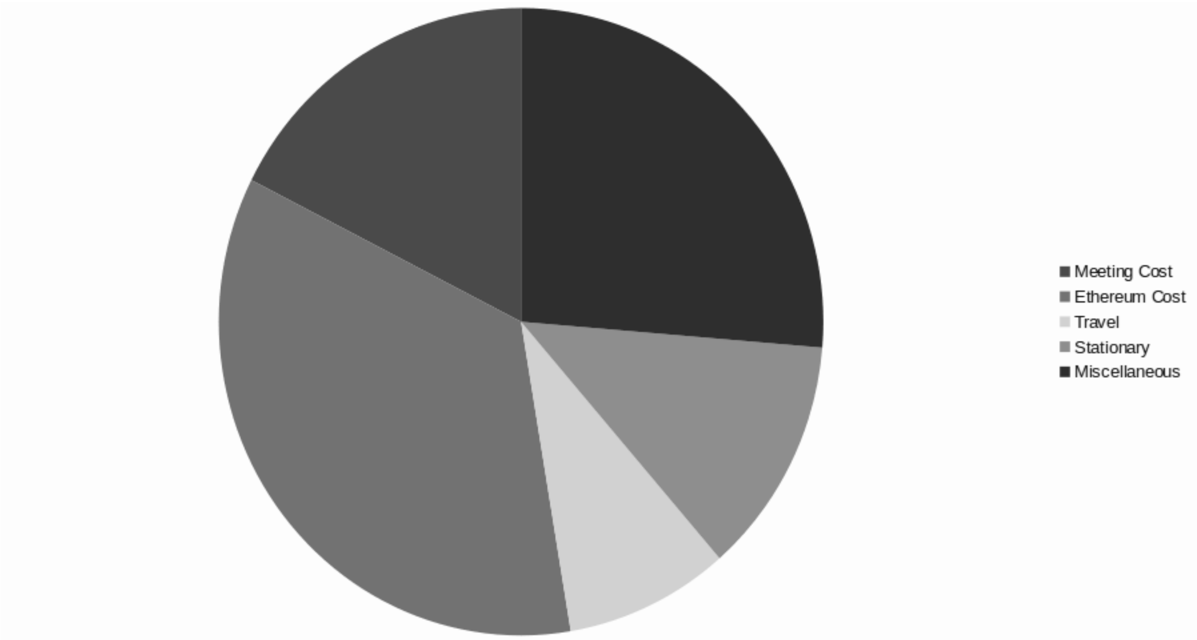


Figure 10: Cost Estimation

# References

[1] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, pp. 6–10, 2016.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] coldfustion, "Why blockchain matters more than you think!" Sep 2017. [Online]. Available: https://www.youtube.com/watch?v=sDNN0uH2Z3o

[4] "The great chain of being sure about things," Oct 2015. [Online]. Available: https://www.economist.com/news/briefing/21677228-technology-behind-bitcoin-lets-people-who-do-not-know-or-trust-each-other-build-dependable

[5] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," *white paper, BigChainDB*, 2016.

[6] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," 2014.

[7] S. Jung, "Filecoin v. sia, storj & maidsafe: The crowded push for decentralized storage," Aug 2017. [Online]. Available: https://medium.com/tokenreport/filecoin-v-sia-storj-maidsafe-the-crowded-push-for-decentralized-storage-7157eb5060c9

[8] "Leveldb - fast and lightweight key/value database from the authors of mapreduce and bigtable - high scalability -." [Online]. Available: http://highscalability.com/blog/2011/8/10/leveldb-fast-and-lightweight-keyvalue-database-from-the-auth.html

[9] [Online]. Available: https://www.sqlite.org/about.html

[10] "Most cited blockchain publications," Nov 2017. [Online]. Available: https://blockchainlibrary.org/2017/10/most-cited-blockchain-publications/