Department of Computer Engineering
Bilkent University

Senior Design Project

Project Short-Name: Nebula

Project Specifications Report

Group Members: *Arda Atacan Ersoy, Kübra Nur Güzel, Seyfullah Yamanoğlu, Başak Şevval Ekici, Serhat Bezmez*

Supervisor: *Eray Tüzün*

Jury Members: *Cevdet Aykanat, Abdullah Ercüment Çiçek*

Innovation Expert: *Duygu Gözde Tümer*

Table of Contents

1.Introduction

Cloud storage has come to rely almost exclusively on large storage providers as trusted third parties to transfer and store data. This system suffers from the inherent weaknesses of a trust-based model. Because client-side encryption is non-standard, the traditional cloud is vulnerable to a variety of security threats, including man-in-the-middle attacks, malware, and application flaws that expose private consumer and corporate data. Moreover, because many storage devices rely on the same infrastructure, failures are correlated across files and systems.

This report delivers the description of Project Nebula. Then insight about the constraints surrounding implementation, economic, ethical, political and sustainability issues regarding Nebula will be given. Afterwards, the professional and ethical responsibilities Nebula brings along will be listed. Finally, the functional and non-functional requirements of Nebula will be provided.

1.1. Description

Nebula is a peer to peer decentralized blockchain based cloud storage system. What does these technical terms mean and why it is an important issue at the moment? Cloud storage becomes more popular everyday but also more people want to keep their data secure and private. With the not pleasant past events from other centralized cloud storage products, such as data losses or hacking to private photographs of celebrities, the idea of Nebula arose. Nebula offers secure and distributed storage of personal data with the use of remote servers for requests and configurations as well as a remote data center infrastructure. Each data will be distributed in an encrypted fashion, with the use of blockchain to make the best possible security services up and running.

Nebula offers many advantages compared to data center-based cloud storage. Data security can be maintained using client-side encryption, while data integrity will be maintained via a proof of storage and retrievability. [1] The impact of infrastructure failures and security breaches will be greatly reduced. An open market for data storage may drive down costs for various storage services by enabling more parties to compete using existing devices since with Nebula users will provide and use the storage themselves. Data on the Nebula network will be resistant to tampering, unauthorized access, and data failures.

Cloud storage has come to rely almost exclusively to be able to provide large storage in order to transfer and store data. The main problem with these systems is that the lack of a trust-based model. Decentralized cloud storage network offers many advantages compared to datacenter-based cloud storage. Nebula will provide the data security with client-side encryption which is not a standard in the systems that are widely used now since they are datacenter-based models.

1.2 Constraints

1.2.1 Implementation Constraints

- GitHub will be the collaboration tool for this project and our informative webpage will be http://nebulaproject.github.io/ , we will also use GitHub as a version control system for the project Nebula.
- We will follow OOP patterns and style of coding.
- For mobile applications, we will use hybrid programming which allows building the same code on multiple platforms.
- Our web, mobile (IOS and Android) and desktop applications will be using the same server-side application.
- Web/mobile and desktop applications will be synchronized.
- We will conduct our development open-source to be able to make it decentralized.
- The system will be based on blockchain infrastructure.

1.2.2 Economic Constraints

- Usage of open source libraries will be free.
- Development of the application relies on personal budget of the project members.

- Github.io pages will be used as a free domain service.
- There will be a payment policy ICO [2] for payments of private network systems for users and companies who want private cloud.
- For Android™ platform, a fee of 25$ needs to be paid so that our application can be published on Google Play™. This payment will be done only once.
- For iOS, a fee of 100$ will need to be paid. This payment must be done every year, so our application can remain published on App Store®.
- System will have mining fee for the miners (doing transactions) of the blockchain.

1.2.3 Sustainability Constraints

- The application will get user feedback for quality enhancement and better business relationships with the customers.
- We will be able to gather anonymous data from users.
- We will know current active users (peers) all the time and guess the current bandwidth and capacity.

1.2.4 Ethical Constraints

- We will abide by the code of ethics of national society of professional engineers [3].
- The user will only be able to access their information after a successful authentication.
- User data on our servers will be encrypted.

1.2.5 Language Constraints

- The interface language of our application will be English.
- For future plans, multi-language support can be implemented.

1.2.6 Security Constraints

- Data provided by our users will be secured by encryption and will not be shared with third parties.

1.2.7 Time Constraints

- The specified reports must be delivered on time also the implementation must be as complete as possible for the demo.

1.3. Professional and Ethical Issues

Since we are developing an application that stores the private data of users, we have to consider about ethical issues that may occur. These problems can be about the security and privacy of the data. During all the states of the development, we will comply with the Code of Ethics outlined by the National Society of Professional Engineers [3].

Since the application will be allowing users to host their hard drive, we are going to have a Nebula Terms of Service that user needs to read before using the system.

In terms of privacy, the user may have some concerns about P2P systems which stores the file of the user to another host peer's storage. We will enforce privacy by both end-to-end and client-side encryption of the file data, so the host would not be able to read it.

We will be using at least one external software libraries while developing our project. Our policy will be firstly search and analysis of the open-source libraries to not get involved with the copyright infringement. If the library we need to use in the project has copyrights, we will use the licensed software library. Any additional source which is added to the application will be properly referenced.

2. Requirements

2.1 Functional Requirements

2.1.1. Server Requirements

- Server should response robust while doing the peer distribution
- Providing private and public keys for each peer in perfection.
- Users ICO wallet information will be kept on server.
- Server has to know each active node and their properties such as bandwidth, storage amount, etc.

2.1.2. Client Applications Requirements

2.1.2.1 - Mobile applications

- Must be efficient in terms of memory and CPU usage
- Users can login with their username and passwords
- They can mine using their user credentials
- Users will be able to see their wallet information
- Users will be able to share their storage from mobile device

2.1.2.2 - Web application

- Users will be able to upload and download files to/from their accounts.
- Web application will have a responsive design to different resolutions of screens.

2.1.2.3 - Desktop application

- The system will store the user’s data by splitting apart, encrypting and distributing it across the decentralized network.

- Users will be able to see their wallet information and let system make mining process.
- Users will be able share their storage from their PC.

2.2. Non-Functional Requirements

2.2.1. Security

- Application should keep the data secure and private via a series of client-side processes before it enters the network. To do that, first the metadata required to remotely verify file integrity should be generated, then the data is split into shards, and each shard should be encrypted.
- End-to-End encryption and client-side encryption will be provided for each user from the network.
- The encrypted data should be then sent out into the network, and the integrity of the data should be checked at regular intervals. So unencrypted data should be never exposed, files in this system should be as secure as the user's key management.
- Program should manage this file storage in terms of blockchain technology and files should be separated into different users. So, no one device has an entire file and all the files are copied on numerous devices, so if a hard drives fails your files won't be lost.
- By the aim of decentralized storage, application should not be vulnerable to hacker attack since it is impossible for anyone without the private keys to decrypt the encoded files.

2.2.2. Usability

- Application should be easy to use for all user types because anyone who can use computer is possible user of this application. Thus, we need to provide user friendly and simple user interface.

- Application should clearly explain what it is aim and why user should use this app. Since its aim and infrastructure is different than the well-known centralized file storage systems.

2.2.3. Cost

- Application should provide lots of hosts for decentralized file storage. Hosts should determine their own prices for renting out their hard drives (similar to the Airbnb model). This ensures that the prices will stay competitive as time goes on.
- Application should provide much less expensive services compared to major players. Application should offer from ⅓ to ⅕ cost of global centralized storage companies. (Per terabyte storage cost, Amazon: 23$, Google Cloud: 20$) [4]

2.2.4. Performance

- Application should enable segmented and distributed files allowance for multi-threaded concurrent downloads.
- The speed of user’s file storage will depend on the performance of local storage who services to our user. In order to compete with centralized system, we need more people to serve as storage disk. As we increase our competition between people who provide disk storage. We can increase our performance.

2.2.5. Extendibility

- Application should be suitable for changes because when users demand new features from us, we should have such infrastructure that can provide it.
- Application should provide the payment method for storage and bandwidth as user use it. User should not be restricted to buy very huge storage files unless user needs it.

2.2.6. Marketability

- In terms of other non-functional requirements, decentralized file storage system has some advantages to centralized ones. This is very important to use this application for the users who concerns these non-functional requirements.

- Big database companies are willing to work with decentralized file storage services such as Couchbase, MongoDb, InfluxDb. [5]

3. References

[1]http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8036727&isnumber=8036702

[2]Momoh, O. (2018). Initial Coin Offering (ICO). Available: investopedia.com/terms/i/initial-coin-offering-ico.asp.[Accessed: 13- Oct- 2018]

[3] "Code of Ethics | National Society of Professional Engineers", Nspe.org, 2018. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 13- Oct- 2018].

[4] https://www.investinblockchain.com/what-is-siacoin/

[5] Decentralized Cloud Storage — Storj. (2018). Available: https://storj.io/. [Accessed: 13- Oct- 2018].