Sri Lanka Institute of Information Technology

VAULT

A shared, distributed, and redundant storage solution

Project ID - 19-002

B.A Ganegoda – IT16016026

Software Requirements Specification - CDAP-I

B.Sc. Special (Honors) Degree in Information Technology

DECLARATION

I, B.A.Ganegoda declare that this is my own work and this software requirement specification document does not incorporate with acknowledge any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

……

B.A.Ganegoda

Table of Contents

Table of Figures

List of Tables

1. Introduction

1.1 purpose

The purpose of this SRS document is to outline the requirements and present a detailed description of the process needed for Identity management, key derivation for file sharing and encryption**.** The document will explain the purpose, features, functional and non-functional requirements, design constraint, project approach, constraint under which above mentioned modules must operate and how the modules will interact with the other modules and the external applications. The information is organized in such a way that the developers and customers will not only understand the boundaries within which they need to work, but also what functionality needs to be developed by the developers and in what order.

1.2 Scope

The component which will be discussed in this document is called "Identity management, key derivation for file sharing and encryption". The components are divided into two important parts. Those are Identity management and the other part is key derivation for file sharing and encryption. The main purpose of this first part, Identity management is to authenticate the user and identify each and every node in the cluster and then identify the secure nodes and rouge nodes if available. Hence for identification, the system uses public and private key encryption along with a master key. This part also covers the integrity check for the files that are stored and encrypted. The main purpose of the key derivation is to handle the public and private key derivations along with the user keys to ensure a correct encryption will be done and files will be shared encrypted and only the intended users will get access. For the encryption, an AES 256 scheme will be used to achieve the maximum speed and to ensure security adhering to the industry standards.

1.1 Definitions, Acronyms, and Abbreviations

Table 1 : Definitions and Abbreviations

OS	Operating System
UI	User Interface
SRS	Software Requirement Specification.

1.4 overview

Remainder of this document mainly can be divided into two sections plus appendix. First section is called overall description and it provides readers an understanding about the overall functionalities of the component, and the interactions of the component with the other components. Future more this section describes functional and nonfunctional requirements, design constraints which includes user interfaces, system interfaces, hardware interfaces and system constraints.

Second section which called specific requirements provides requirements specifications in a detailed manner. Specify requirements clearly for different audiences to understand by using various kinds of specification techniques. Future more software system attributes and performance requirements will be discussed in this section.

2 Overall Descriptions

Identity management, key derivation for file sharing and encryption

Above mentioned two functions will be discussed in this document; Identity management, key derivation for file sharing and encryption. Node authentication by public key cryptography can be used to identify the trusted nodes in the cluster. Every user has a list of public keys of every node stored in the custom blockchain. A Master key(M) will be sent to validate the nodes that is connected in the network. By comparing the master key, the nodes will be able to verify the secure nodes. The encryption will be done using AES 256 for maximum speed and security. The hash values that are stored in the blockchain will provide a validation of the files or file parts that are stored in the cluster and ensure the integrity of the files. When considering the sharing scenario, a link will be shared with the intended user that has the locations of the file parts that are stored in the network. With that link, the user can access the files by downloading the file into the storage. A separate temporary key will be used to encrypt the file. That key will be shared with the trusted node using Public-key cryptography.

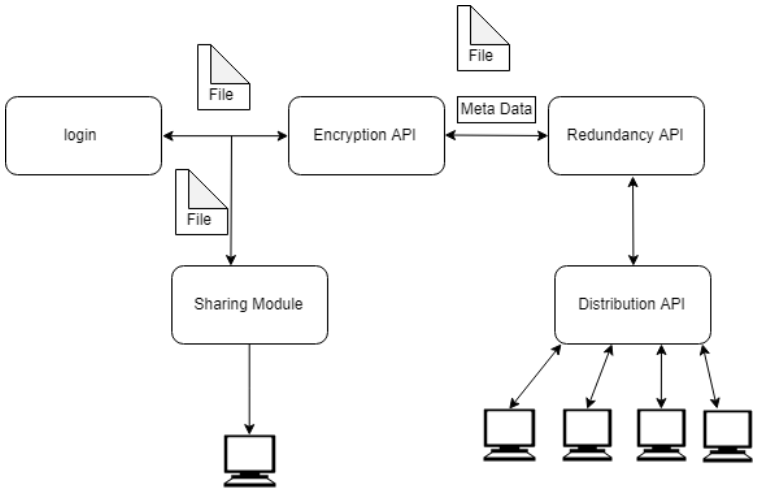

Figure 1: Identity management, key derivation for file sharing and encryption overview

2.1 Product perspective

Below diagram will describe the similarities and the differences of the final product, not only the component mentioned in this document, with the existing products.

Table 2 Feature Comparison

Features	MooseFS	IBM Spectrum Scale	OCFS 2	OrangeFS	BWFS	Minio	Ceph	VAULT
High Availability	✓	✓	✓	✓		✓	✓	✓
Scalability	✓	✓	✓	✓	✓	✓	✓	✓
Minimal Investment	✓			✓		✓	✓	✓
Big Data Support	✓	✓	✓	✓		✓	✓	✓
Data encryption		✓				✓	✓	✓
Data Recovery	✓	✓		✓	✓	✓	✓	✓
Platform independent	✓	✓						✓
Security		✓	✓	✓		✓	✓	✓
Data Redundancy					✓	✓		✓
Minimum additional storage for backup								✓
Blockchain integration								✓
Easy to setup and run								✓
File Sharing								✓

2.1.1 System interfaces

Since the system will be designed and developed according to the module-based approach each research component will be designed, developed and tested out individually since they all are developed as modules which then can be imported and assemble the complete software. Since NodeJS will be used as the developing language application can be run inside any Linux, MacOS and Windows environments (Platform independent). Since the final product and the component mentioned in this document both acts as standalone solutions final product, or the component mentioned in this document won't be interacting with the other existing applications other than the web browser which will be needed to display the user interfaces of the web-application. Although hardware monitoring function will be interacting with the

Operating Systems build in features such as Self-Monitoring, Analysis, and Reporting Technology to gather hardware health related data.

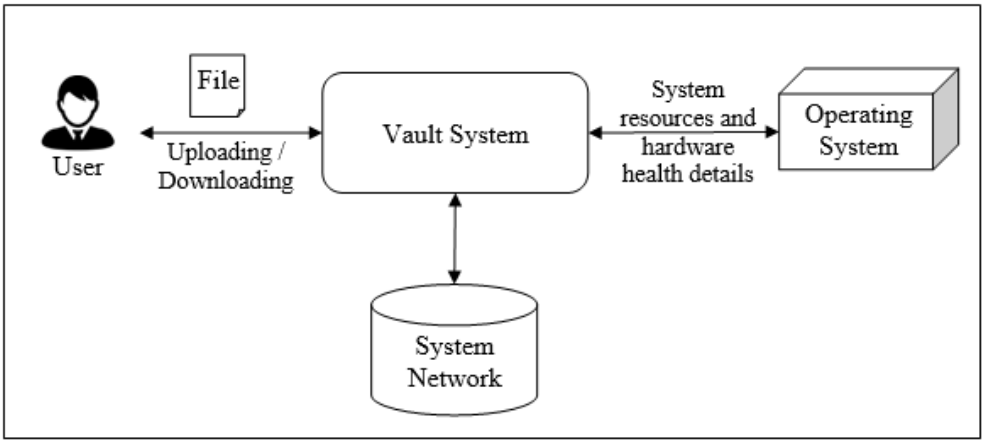

Figure 2:System Overview

2.1.2 User interfaces

Research component described in this document primarily can be divided into two specific functions Identity management, key derivation for file sharing and encryption, as mentioned before in this document. Since identity management consists of the user authentication process, node authentication and the process of file integrity check, these modules excluding the user authentication, it doesn't requires any specific interface. This module takes data at beginning and then encrypts it. And then validate the nodes in the process. Interface which will be used to share the files, encrypt the files and file integrity using hashing will be included in the section 3.

User authenticate will enable to validate the users who are valid and give access to those users only. A user can log on to any computer and can sign-in to their account at any given moment.

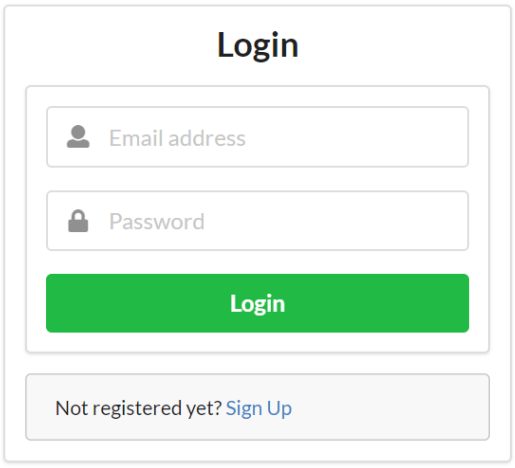

Figure 3: Login Interface

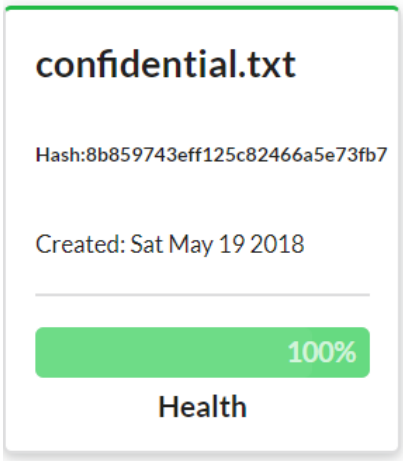

Figure 4 : Integrity check with hash values

2.1.3 Software interfaces

Since the application developed using module-based approach, the component mentioned in this document was mainly developed as two modules, Identity management, key derivation for file sharing and encryption, hence both of those mentioned modules act as middle layer processes they will be only store the public keys of the trusted nodes on the custom blockchain. And it won't be storing any permanent data or else won't be connecting with external applications. Only requirement needs to run the component successfully is a physical machine with NodeJS installed on them (Platform Independent).

2.1.4 Communication interfaces

Both modules as mentioned in the above section act as middle layer processes, hence they won't be needing any external communication interfaces other than the interfaces needed to communicate with the other modules within the application. Encryption handling module will need two specific internal interfaces to connect with the Redundancy module and the distribution module. The sharing module will be connected to the Blockchain API to receive the file and node details.

2.1.5 Memory constraints

Since the application is engineered to use disk space more than Memory, a machine with at least 2GB is sufficient.

2.1.6 Operations

Prior to use, Docker application will be deployed among the users by using the internal network or else physically. Application basically have two types of user accounts called administrator accounts and normal accounts.

All the user account related activities will be carried out by the administrative users using the web application through web browsers. Administrators have the ability to connect/ disconnect workstations from the system, can view available workstations at a certain time and can change the master key for the nodes.

When a normal user accesses the web application from the web browser a logging page will be displayed to the user, and users can use the logging credentials provided by the administrators to log in to the application. When a normal user logs in they can either read, upload, download and also share files with the nodes in the network. The files that are uploaded to the system will be stored by partitioning the files among the nodes in the network and these files are stored in a secure manner.

2.1.7 Site adaptation requirements

In order to run the application users must complete few tasks in a certain order which will be presented below in the order that users must carry them out. When users accessing the application, they will be guided using English language.

1. NodeJS must be installed
2. Docker application which contains the system must be deployed
3. All the workstations must connect to the local network.
4. At least there must be 6-10 users that can contribute 10 GB storage space by each
5. Web browser must be installed
6. Users must have login credentials to access the web application

2.2 Product functions

- Since the above mentioned two modules include most functions that are not user operated, those were not included in the use case diagram.

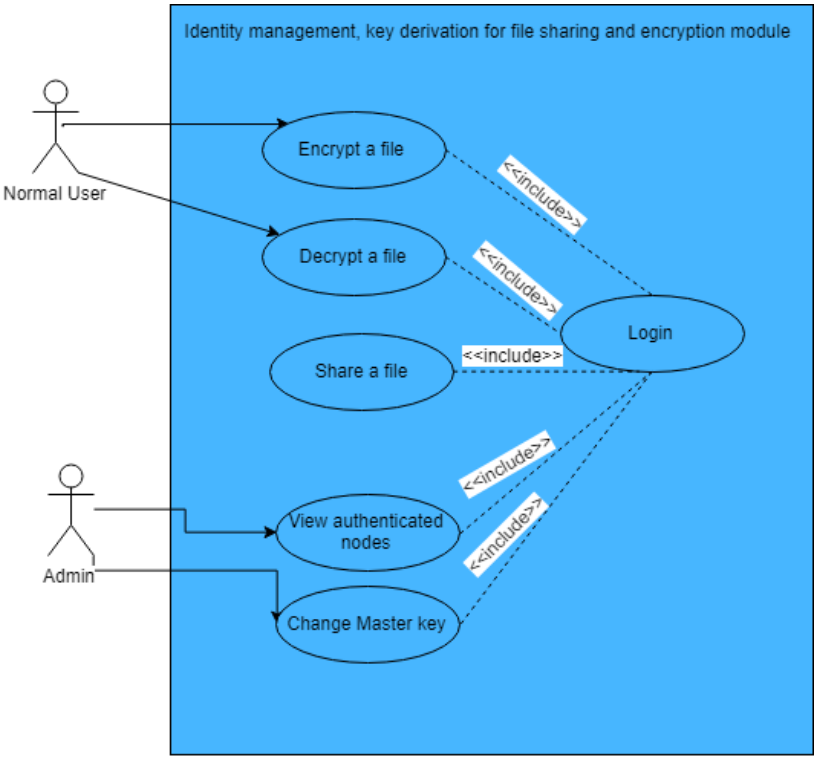

Figure 5 : Identity management, key derivation for file sharing and encryption use case diagram

Login

Table 3 Login use case scenario

Use Case No	01
Use Case	Login
Actors	Normal User, Administrator
Pre-Conditions	User must be a registered user
Flow of Event	1. Enter the username and password 2. Click Login
Post Conditions	Allow user to the web application
Alternatives	Display an error message to inform that the user credentials are invalid.

Register new user

Table 4 New user registration use case scenario

Use Case No	02
Use Case	Register new user
Actors	Normal User
Pre-Conditions	User must have access to the system
Flow of Event	1. Press the "register" button 2. Input new user name 3. Input new password 4. Press "submit" button 5. Web application will display message "Registration Successful"
Post Conditions	Display "Registration Successful" message
Alternatives	Display an error message to inform user that the new user cannot be registered.

View authorized nodes

Table 5 View authorized nodes use case scenario

Use Case No	03
Use Case	View authorized nodes
Actors	Administrator
Pre-Conditions	User must be logged in
Flow of Event	1. Select "nodes" button 2. Select authorized nodes option. 3. View all authorized nodes on the cluster.
Post Conditions	Display authorized nodes in a separate viewer
Alternatives	Display an error message to inform user that the nodes will not be displayed.

Share a files

Table 6 Share a file use case scenario

Use Case No	04
Use Case	Share a file
Actors	Administrator
Pre-Conditions	User must be logged in
Flow of Event	1. Select a file from the file directory. 2. Press on "Share" button 3. Type in the temporary password to encrypt the file. 4. Press on "Select recipient" button. 5. Select the intendent recipient 6. Press "Ok" button to send 7. Web application will display message "Link shared with the recipient"
Post Conditions	Web application will display message "Link shared with the recipient" and a link will be shared with the intendent recipient
Alternatives	Display an error message if there are any complications in sharing the link or selecting the user.

Change Master Key

Table 7 Change Master Key case scenario

Use Case No	05
Use Case	Change Master Key
Actors	Administrator
Pre-Conditions	User must be logged in
Flow of Event	1. Select change master key option from the menu. 2. Type the new password 3. Press on "Change" button 4. Confirm the change by pressing the "Ok" button. 5. Web application will display message "Password changed"
Post Conditions	Web application will display message "Password changed"
Alternatives	Display an error message if there are any complications in when changing the password.

2.3 User characteristics

There can be mainly two types of users in the system administrators and normal users.

Administrator- Software/hardware professional who will be configuring the system and maintain the consistency of the system.

Normal Users- Anyone with the basic knowledge of computing who will be using the system to store their files and download them when needed.

2.4 Constraints

Identity management module constraints

- NodeJS must be installed

key derivation for file sharing and encryption module constraint

- NodeJS must be installed

2.5 Assumptions and dependencies

It was assumed that the nodes are connected to a secure Network and There was no other assumption made due to the reason both modules are platform independent and require no additional assist from third party applications or any additional hardware.

3 Specific requirements [1] (for “Object Oriented” products)

3.1 External interface requirements

Table 8 File Sharing Interface description

Name of item	File Sharing Interface
Description of purpose	User will be able to share the files of his or her that are stored with intended parties with this interface.
Source of input or destination of output	Intended node address from the blockchain
Valid range, accuracy and/or tolerance	80%
Units of measure	-
Timing	Depends on the file size
Relationships to other inputs/outputs	Availability information will be outputted to the distribution API when segments of a file is recollecting.
Screen formats/organization	Screen is organized in a monitor view
Window formats/organization	-
Data formats	-

3.1.1 User interfaces

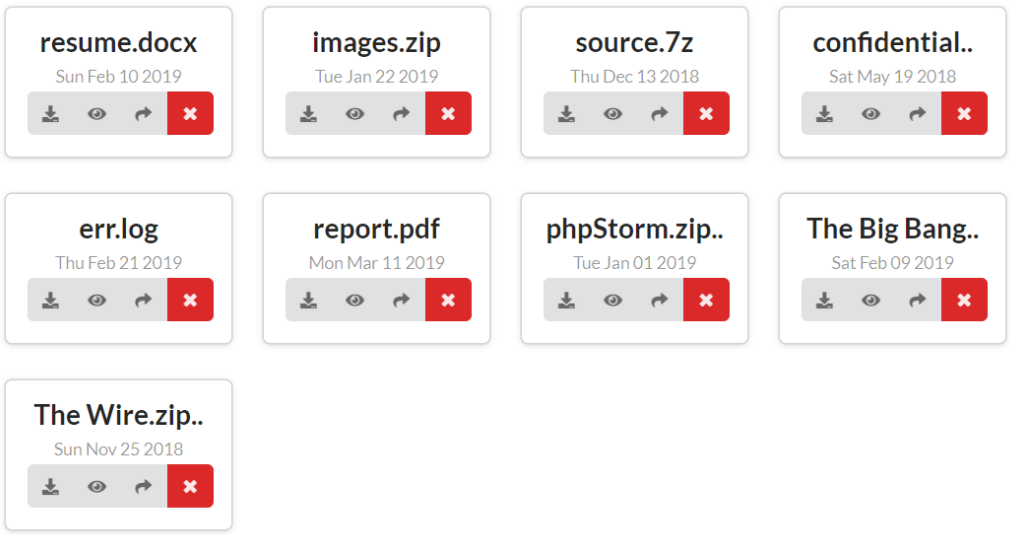

Figure 6 : File Sharing Interface

3.1.2 Hardware interfaces

Since the software does not directly interact with Hardware level components and doesn't

rely on how storage devices handle data, there is no requirement to document hardware interfaces.

3.1.3 Software interfaces

Other than NodeJS and Operating System the research component mentioned in this document won't be interacting with any other software application.

3.1.4 Communication interfaces

Only the identity management will need an external interface to authenticate the users and other than that, other parts of the identity management and the key derivation for file sharing and encryption module acts as a middle layer processes, hence they won't be needing any external communication interfaces other than the interfaces needed to communicate with the other modules within the application and the operating system.

Table 9 Identity management module communication interfaces

Interface	External Module	Connection details
Interface 1	Login module	When user tries to log into the system the user is verified using a simple username and password basis authentication and this module can enable creation of users as well. Multiple failed login attempts will disable a user access from the system.
Interface 2	Common Module	When a file is segmented and ready to send to nodes for storing, this module will be used to verify the trusted nodes in the cluster

Table 10 the Key derivation for file sharing and encryption module communication interfaces

Interface	External Module/ Feature	Connection details
Interface 1	Redundancy Module	Gather disk health and availability information from all over the network through distribution function.

3.2 Classes/Objects

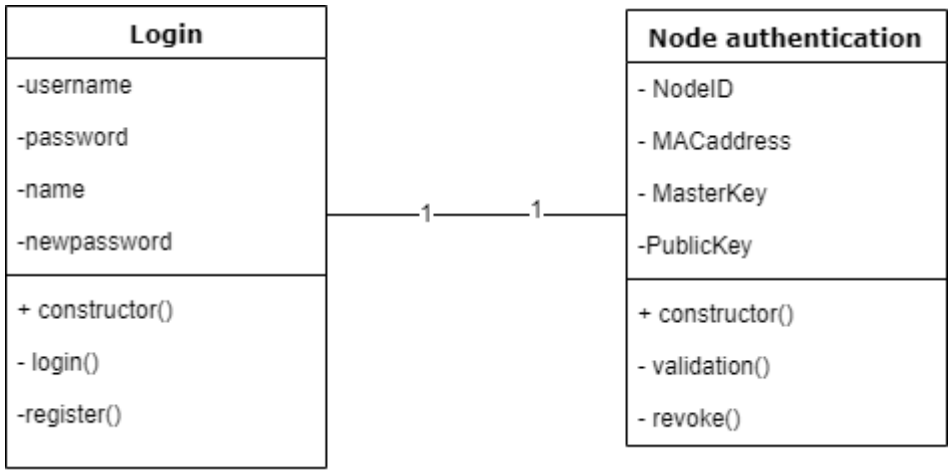

Figure 7: Identity management module class diagram

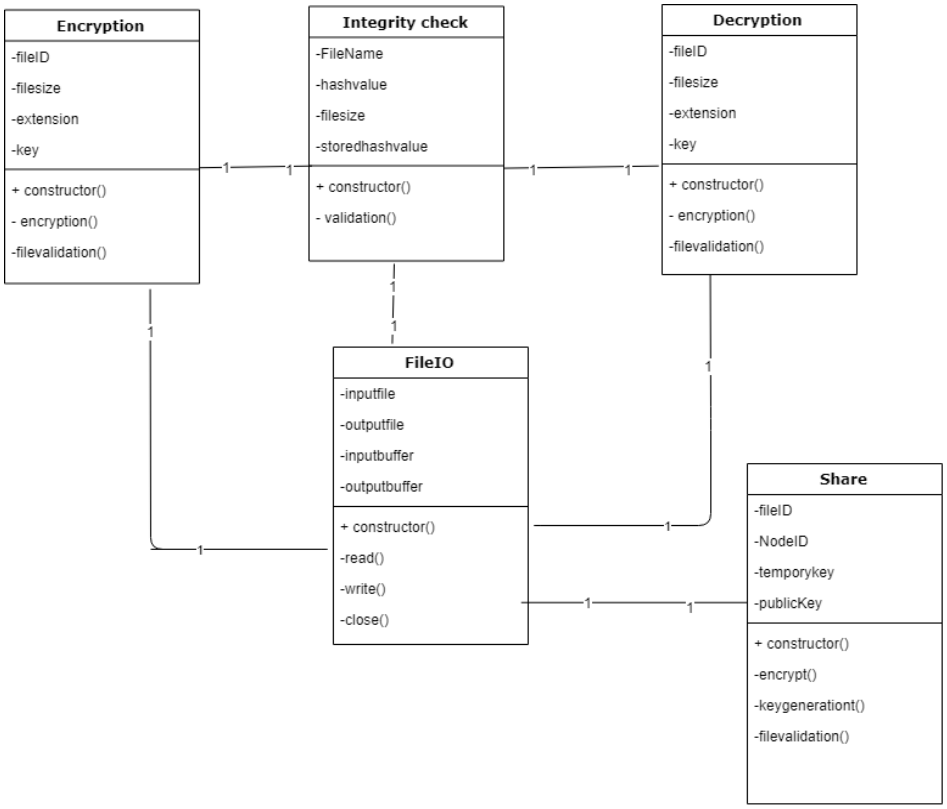

Figure 8: Key derivation for file sharing and encryption module class diagram

3.3 Performance requirements

Storage and network requirements can be identified for the blockchain and Messenger API facilitation. The host machine should have the necessary capacity to host storage for the Rethink DB server. The network should be reliable. Network speed/bandwidth is not a main concern since the amount of data throughput required by the module is very low.

3.4 Design constraints

There are no specific design constraints involves with the component which is mentioned above in this document, as long as the interfaces are easy to understand and use.

3.5 Software system attributes

In this section the features which will be offered to the customers will be described.

3.5.1 Reliability

Ability of a system to maintain its ordinary operation within given time in a given environment with minimum failures is called reliability. Purpose of the Identity management and the encryption module is to increase the Reliability.

As mentioned earlier Identity management module will identify the secure nodes that are in the cluster, thus giving a reliability to the network and the hash value comparing will ensure that the files that are stored will make it tamperproof and will ensure the integrity of the files

The encryption module will increase the reliability by allowing to encrypt the data, thus making sure that the data is segmented and stored more securely. administrative users to continuously monitor the system. By monitoring the disk health administrators can move the data inside one disk to another disk if that certain disk's health is not in good condition, which allows them to stop future failure.

3.5.2 Availability

Probability of a system functioning when its services are required by the users of the system is called availability of a system. Another purpose of the Identity management, key derivation for file sharing and encryption module is to maintain the availability of the system.

Identity management module will allow a user to quickly store their data in secure nodes, this is done by quickly identifying nodes that are secure.

The encryption module will provide a clean and smooth encryption and a decryption of a file maintaining the availability of the system.

3.5.3 Security

Security of a system is the function which allows system to provide its services to its legitimate users, while resisting the other unauthored users from gaining access to the system or its data and resisting authorized

users from performing unauthorized actions Identity management, key derivation for file sharing and encryption module plays a major roles in maintaining security process.

Identity management will very verify the users, authenticate the trusted node. key derivation for file sharing and encryption module mainly focusses on encrypting the files, this way the files are secured in a secured manner and sharing part will enable secure sharing.

3.5.4 Maintainability

Maintainability is the ability to change the systems functionalities and increase the performance by applying system repairs and updates while maintaining systems availability, security and reliability. Apart from the details mentioned in this document major changes to the system cannot be expected in the Identity management, key derivation for file sharing and encryption modules, but there can few changes in the class structure and the interfaces which was depicts in this document.

4 Supporting information

4.1 References

[1] BlueWhale, "bwstor.com," BlueWhale, 03 2019. [Online]. Available: www.bwstor.com.cn/templates/T_product_EN/index.aspx?nodeid=150&page=ContentPage&contentid=402. [Accessed 02 03 2019].

[2] Ceph, "ceph.com," Ceph, 03 2019. [Online]. Available: https://ceph.com/ceph-storage/file-system/. [Accessed 01 03 2019].

[3] Minio, "minio," Minio, 02 2019. [Online]. Available: https://www.minio.io/. [Accessed 03 2019].

[4] Oracle, "oracle.com," Oracle, 03 2019. [Online]. Available: https://oss.oracle.com/projects/ocfs2/. [Accessed 03 2019].

[5] The OrangeFS Project, "orangefs," The OrangeFS Project, 03 2019. [Online]. Available: http://www.orangefs.org/. [Accessed 03 2019].

[6] IBM, "www.ibm.com," IBM, 03 2019. [Online]. Available: https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage. [Accessed 03 2019].

[7] "moosefs.com," moosefs, [Online]. Available: https://moosefs.com/.