UNIVERSITY OF BEIRA INTERIOR
Engineering

Secure and Efficient Storage of Multimedia Content in Public Cloud Environments Using Joint Compression and Encryption

André Filipe Prata Ferreira

Dissertation for obtaining the degree of Master of Science in
Computer Science and Engineering
(2nd Cycle Studies)

Advisor: Prof. Dr. Mário Marques Freire

Covilhã, October 2013

Dissertation prepared at University of Beira Interior and at Instituto de Telecomunicações, within Multimedia Signal Processing – Covilhã Group, and submitted for defense in a public examination session at University of Beira Interior.

Work partially financed by the Portuguese Science and Technology Foundation within the Strategic Project PEst-OE/EEI/LA0008/2013.

Dedicatory

I dedicate this work to the most important people in my life.
For what they taught and transmitted me, for the unconditional and unceasing support.

To my family.

Acknowledgments

This work is the culmination of a process of learning and personal development, which would not have been possible without the support and dedication of some people to whom I express my gratitude. A heartfelt thanks:

To my advisor, Professor Mário Marques Freire, for the views, the availability, scientific rigor and interest throughout this work.

To all my teachers who encouraged me in the pursuit of knowledge and given me the foundation necessary to develop and implement this research.

To all my friends for unforgettable moments of friendship.

To my mother Célia Prata, who accompanied me, helped and encouraged, providing me the tools necessary to achieve this important goal. Teaching me everyday to never give up on my goals.

To my father Armando Ferreira that helped me to overcome the obstacles in this work.

To my brother David by endless friendship, being always present with a friendly word for another "push".

Foreword

Cloud computing is the latest revolution in information technology that is intrinsically related to the economy. The growing number of applications and the amount of data that need to be managed has made data centers to become an important and essential investment and public cloud computing seems to be a means to control these costs. It follows the needs of technological information and responds to requests from organizations worldwide. The cloud services are crucial for business, offer new products and services to market, allow the storage and processing of data, increasing the agility, efficiency and productivity.

This computing architecture extends its services to the world market. Providers offer services that run in the cloud and can be accessed using the Internet Protocol, regardless of location. Cloud services are scalable, the resources required of storage and computational power can be increased or decreased based on customer needs and have many guests, they provide hosting service and safe for multiple and simultaneous clients using the same cloud infrastructure resources.

The benefits of cloud computing to business are obvious, but many organizations are still hesitant to use the cloud for security reasons. Cite reliability concerns, putting safety as a barrier. Concerns range from the protection of confidential business information to compliance with legal requirements and the protection of personal data. However, at present, there are solutions available that provide a reliable cloud. The implementation of public and private clouds have an appropriate set of important principles in order to ensure safety to users and customers a trusted environment of cloud computing.

Resumo

A Computação em nuvem é um paradigma ainda com muitas áreas por explorar que vão desde a componente tecnológica à definição de novos modelos de negócio, mas que está a revolucionar a forma como projetamos, implementamos e gerimos toda a infraestrutura da tecnologia da informação.

A Infraestrutura como Serviço representa a disponibilização da infraestrutura computacional, tipicamente um datacenter virtual, juntamente com um conjunto de APIs que permitirá que aplicações, de forma automática, possam controlar os recursos que pretendem utilizar. A escolha do fornecedor de serviços e a forma como este aplica o seu modelo de negócio poderão determinar um maior ou menor custo na operacionalização e manutenção das aplicações junto dos fornecedores.

Neste sentido, esta dissertação propôs-se efetuar uma revisão bibliográfica sobre a temática da Computação em nuvem, a transmissão e o armazenamento seguro de conteúdos multimédia, utilizando a compressão sem perdas, em ambientes em nuvem públicos, e implementar um sistema deste tipo através da construção de uma aplicação que faz a gestão dos dados em ambientes de nuvem pública (dropbox e meocloud).

Foi construída uma aplicação no decorrer desta dissertação que vai de encontro aos objectivos definidos. Este sistema fornece ao utilizador uma variada gama de funções de gestão de dados em ambientes de nuvem pública, para isso o utilizador tem apenas que realizar o login no sistema com as suas credenciais, após a realização de login, através do protocolo Oauth 1.0 (protocolo de autorização) é gerado um token de acesso, este token só é gerado com o consentimento do utilizador e permite que a aplicação tenha acesso aos dados/ficheiros do utilizador sem que seja necessário utilizar as credenciais. Com este token a aplicação pode agora operar e disponibilizar todo o potencial das suas funções. Com esta aplicação é também disponibilizado ao utilizador funções de compressão e encriptação de modo a que possa usufruir ao máximo do seu sistema de armazenamento cloud com segurança. A função de compressão funciona utilizando o algoritmo de compressão LZMA sendo apenas necessário que o utilizador escolha os ficheiros a comprimir. Relativamente à cifragem utilizamos o algoritmo AES (Advanced Encryption Standard) que funciona com uma chave simétrica de 128bits definida pelo utilizador.

Alicerçámos a investigação em duas partes distintas e complementares: a primeira parte é composta pela fundamentação teórica e a segunda parte consiste no desenvolvimento da aplicação informática em que os dados são geridos, comprimidos, armazenados, transmitidos em vários ambientes de computação em nuvem. A fundamentação teórica encontra-se organizada em dois capítulos, o capítulo 2 - "Background on Cloud Storage" e o capítulo 3 "Data Compression".

Procurámos, através da fundamentação teórica, demonstrar a pertinência da investigação, transmitir algumas das teorias pertinentes e introduzir, sempre que possível, investigações existentes na área. A segunda parte do trabalho foi dedicada ao desenvolvimento da aplicação em ambiente "cloud". Evidenciámos o modo como gerámos a aplicação, apresentámos as funcionalidades, as vantagens. Por fim, refletimos sobre os resultados, de acordo com o enquadramento teórico efetuado na primeira parte e o desenvolvimento da plataforma.

Pensamos que o trabalho obtido é positivo e que se enquadra nos objetivos que nos propusemos atingir. Este trabalho de investigação apresenta algumas limitações, consideramos que o tempo para a sua execução foi escasso e a implementação da plataforma poderia beneficiar com a implementação de outras funcionalidades. Em investigações futuras seria pertinente dar

continuidade ao projeto ampliando as potencialidades da aplicação, testar o funcionamento com outros utilizadores e efetuar testes comparativos.

Keywords

Cloud computing, Lossless Compression, Cloud Security,

Resumo alargado

A computação em nuvem insere-se no meio informático como a última revolução na tecnologia da informação do século XXI, deslocando todos os dados e aplicações dos utilizadores para grandes centros de armazenamento, distribuídos na forma de serviços baseados na Internet, acelerando a disponibilização de novos produtos e serviços para o mercado e aumentando a eficiência e a utilização de recursos.

Dado o crescimento exponencial de dados que precisam de ser transmitidos ou armazenados e a crescente utilização de sistemas ligados em redes, o desenvolvimento de aplicações que utilizam dados multimédia (texto, som, imagem e vídeo) que ocupam muito espaço e dificultam a sua manipulação, tornam a compressão de dados de extrema importância e cada vez mais necessária, sendo necessário investir cada vez mais em melhores tecnologias de transmissão e armazenamento. Os métodos de compressão podem ser divididos em duas categorias distintas, algoritmos de compressão sem perdas em que o resultado final é igual ao inicial e o algoritmo de compressão com perdas em que a representação dos dados finais é ligeiramente diferente dos dados iniciais. As técnicas de compressão sem perdas são necessárias em certas aplicações em que a exatidão da informação é essencial, os ficheiros não podem sofrer alterações devido ao processo de compressão. O desempenho de um esquema de compressão pode ser obtido de várias formas, sendo a forma mais comum obtida dividindo o espaço de armazenamento ocupado pelo fluxo de dados originais, pelo espaço consumido pelo fluxo de dados comprimidos ou de códigos. Quanto maior for o rácio de compressão, menor será o comprimento do fluxo de dados comprimidos e menor o espaço de armazenamento consumido pelo fluxo de dados comprimidos, ou menor será a largura de banda necessária para a sua transmissão numa rede de computadores. O armazenamento e processamento de dados é efetuado em serviços virtuais que podem ser acedidos de qualquer lugar do mundo, a qualquer hora, não havendo necessidade de instalação de nenhum programa específico.

A utilização da Computação em nuvem traz diversas vantagens que aliciam os utilizadores com a agilização dos procedimentos:

- Para utilizar um aplicativo em nuvem, não é necessário proceder à adequação do sistema operacional e ao hardware do computador,

- Sem que o utilizador se preocupe ou tenha sequer conhecimento, as atualizações dos softwares em Computação em nuvem são efetuadas de forma automática,

- A tecnologia em computação em nuvem é muito vantajosa para trabalhos corporativos, já que o compartilhamento de informações e arquivos está alojado num local comum permitindo que os utilizadores acedam e utilizem, sem terem que trabalhar em diferentes versões,

- Os dados armazenados nos softwares nas nuvens podem ser acedidos em qualquer lugar, bastando que haja acesso à Internet, evitando os riscos de perda de informações armazenadas localmente,

- Menor custo de manutenção da infraestrutura física e de hardware de redes locais cliente/servidor.

A escolha de entre os diferentes modelos de infraestrutura é ponderada pelos utilizadores. Existem nuvens públicas, privadas e hibridas, com benefícios complementares e cuja opção na implementação deverá equacionar um conjunto de questões. As nuvens públicas estão disponíveis para qualquer utilizador, a partir de um acesso à Internet, podendo este aceder e utilizar de forma quase imediata, as nuvens privadas estão normalmente localizadas dentro de uma infraestrutura privada sob controlo de uma organização. Os utilizadores podem

optar pelo modelo mais conveniente para responder às suas necessidades, tendo em conta um conjunto de fatores tais como: o investimento inicial, o volume de informação, a longevidade dos dados, o desempenho, os padrões de acesso e a localização, a segurança e o isolamento da informação, a confidencialidade e a destruição da informação, os acordos de nível de serviço e os recursos técnicos próprios.

A segurança é uma das principais preocupações para as empresas interessadas em implantar uma estratégia de computação em nuvem. Apesar dos benefícios, os utilizadores ainda hesitam em adotar a computação em nuvem colocando a segurança como uma barreira. Referem-se a riscos que podem afetar os serviços de computação em nuvem como a indisponibilidade, a privacidade, o suporte, o aprisionamento e interoperabilidade e a conformidade. A segurança é uma preocupação prioritária e tem uma parcela de influência nas decisões. Este comportamento dos utilizadores deve ser visto pelos provedores de serviços em nuvem como um importante fator para o aprimoramento das suas práticas de segurança.

Para permitir um armazenamento e uma gestão confiável de dados, existem requisitos básicos que devem garantir a segurança da informação: A confidencialidade que assegura que a informação apenas pode ser acedida por pessoas autorizadas, a integridade da informação pois esta deve ser recuperada na sua forma original e a disponibilidade, o sistema tem de estar sempre disponível, mesmo na ocorrência de falhas. As ameaças podem ser diversas (passivas, ativas, maliciosas, não maliciosas...). Para lidar com elas, torna-se necessário a definição de políticas e mecanismos de segurança visando dar suporte à prevenção para evitar que invasores violem os mecanismos de segurança, à deteção que é a habilidade de detetar invasão aos mecanismos de segurança e à recuperação que consiste no mecanismo para interromper a ameaça, avaliar e reparar danos, além de manter a operacionalidade do sistema caso ocorra invasão ao sistema.

Na segurança da informação existem mecanismos para preservar a informação, de forma a garantir a sua disponibilidade, confidencialidade, integridade e autenticidade, estes mecanismos são designados por mecanismos de controlo às ameaças e consistem nos seguintes procedimentos:

- O controlo de acesso que permite controlar as pessoas que estão autorizadas a entrar em determinado local e regista o dia e hora de acesso, controlando e decidindo as permissões que cada utilizador tem. Um sistema de controlo de acesso é constituído por diferentes equipamentos periféricos de controlo e comando, interligados a uma única unidade de controlo que permite, em diferentes pontos, vários acessos.

- O sistema de deteção de intrusos que alerta os administradores para a entrada de possíveis intrusos nos sistemas. Estes sistemas tentam reconhecer um comportamento/ação intrusiva, através da análise das informações disponíveis num sistema de computação ou rede.

- A criptografia que é a arte de codificação e permite a transformação reversível da informação de forma a torná-la inteligível a terceiros.

No entanto mesmo com estas medidas de proteção, muitas pessoas acreditam que a informação armazenada num sistema de armazenamento remoto é vulnerável. Existe sempre a possibilidade de algum modo, de ter acesso à informação. Existe também a possibilidade de funcionários da empresa com acesso aos servidores poderem roubar, alterar ou destruir informação. As empresas no negócio de armazenamento em nuvem investem muito dinheiro em medidas de segurança para limitar a possibilidade de roubo ou corrupção da informação. Para responder às necessidades de segurança, os utilizadores têm uma grande variedade de opções e devem procurar que os seus prestadores estejam certificados. Alguns fornecedores de serviços de computação em nuvem oferecem serviços de encriptação integrados nas suas ofertas de ar-

mazenamento e existe uma gama de aplicações a quem os clientes podem comprar a prestação de serviços de cifragem, proteção a ataques de negação de serviço e medidas de controlo de acesso especificamente adaptados a implantações em computação em nuvem.

De acordo com a revisão bibliográfica efetuada acerca da temática computação em nuvem, transmissão e armazenamento seguro de conteúdos multimédia, utilizando a compressão sem perdas, em ambientes de nuvem pública, alicerçámos a investigação em duas partes distintas e complementares, a primeira parte, está organizada em dois capítulos, o capítulo 2 - "Background on Cloud Storage" e o capítulo 3 - "Data Compression". A segunda parte consiste no desenvolvimento de uma aplicação informática em que os dados são geridos, comprimidos, armazenados e transmitidos em vários ambientes de computação em nuvem. Esta aplicação fornece ao utilizador uma variada gama de funções de gestão de dados em ambientes de nuvem pública, em que para isso o utilizador tem apenas que realizar o login no sistema com as suas credenciais. Após a realização do login, através do protocolo Oauth 1.0 (protocolo de autorização) é gerado um token de acesso, em que este token só é gerado com o consentimento do utilizador e permite que a aplicação tenha acesso aos dados/ficheiros do utilizador sem que seja necessário utilizar as credenciais. Com este token a aplicação pode agora operar e disponibilizar todo o potencial das suas funções. Com esta aplicação é também disponibilizado ao utilizador funções de compressão sem perdas e encriptação de modo a que possa usufruir ao máximo do seu sistema de armazenamento cloud com segurança. A função de compressão funciona utilizando o algoritmo de compressão LZMA sendo apenas necessário que o utilizador escolha os ficheiros a comprimir. Relativamente à cifragem utilizamos o algoritmo AES (Advanced Encryption Standard) que funciona com uma chave simétrica de 128bits definida pelo utilizador. De forma a validar a aplicação, efetuámos uma conjunto de testes com imagens e ficheiros de som tendo obtido resultados satisfatórios, ao nível da compressão sem perdas e da encriptação. Pensamos que o trabalho obtido é positivo e que se enquadra nos objetivos que nos propusemos atingir.

Este trabalho apresenta algumas limitações, consideramos que o tempo para a sua execução foi escasso e a aplicação poderia beneficiar com a implementação de outras funcionalidades, tal como implementação de miagração. Em investigações futuras seria pertinente dar continuidade ao projeto ampliando as potencialidades da aplicação, testar o funcionamento com outros utilizadores e efetuar mais testes comparativos.

Abstract

The Cloud Computing is a paradigm still with many unexplored areas ranging from the technological component to the definition of new business models, but that is revolutionizing the way we design, implement and manage the entire infrastructure of information technology.

The Infrastructure as a Service is the delivery of computing infrastructure, typically a virtual data center, along with a set of APIs that allow applications, in an automatic way, can control the resources they wish to use. The choice of the service provider and how it applies to their business model may lead to higher or lower cost in the operation and maintenance of applications near the suppliers.

In this sense, this work proposed to carry out a literature review on the topic of Cloud Computing, secure storage and transmission of multimedia content, using lossless compression, in public cloud environments, and implement this system by building an application that manages data in public cloud environments (dropbox and meocloud).

An application was built during this dissertation that meets the objectives set. This system provides the user a wide range of functions of data management in public cloud environments, for that the user only have to login to the system with his/her credentials, after performing the login, through the Oauth 1.0 protocol (authorization protocol) is generated an access token, this token is generated only with the consent of the user and allows the application to get access to data/user files without having to use credentials. With this token the framework can now operate and unlock the full potential of its functions. With this application is also available to the user functions of compression and encryption so that user can make the most of his/her cloud storage system securely. The compression function works using the compression algorithm LZMA being only necessary for the user to choose the files to be compressed. Relatively to encryption it will be used the encryption algorithm AES (Advanced Encryption Standard) that works with a 128 bit symmetric key defined by user.

We build the research into two distinct and complementary parts: The first part consists of the theoretical foundation and the second part is the development of computer application where the data is managed, compressed, stored, transmitted in various environments of cloud computing. The theoretical framework is organized into two chapters, chapter 2 - Background on Cloud Storage and chapter 3 - Data compression.

Sought through theoretical foundation demonstrate the relevance of the research, convey some of the pertinent theories and input whenever possible, research in the area. The second part of the work was devoted to the development of the application in cloud environment. We showed how we generated the application, presented the features, advantages, and safety standards for the data. Finally, we reflect on the results, according to the theoretical framework made in the first part and platform development.

We think that the work obtained is positive and that fits the goals we set ourselves to achieve. This research has some limitations, we believe that the time for completion was scarce and the implementation of the platform could benefit from the implementation of other features.In future research it would be appropriate to continue the project expanding the capabilities of the application, test the operation with other users and make comparative tests.

Keywords

Cloud computing, Lossless Compression, Cloud Security,

Contents

List of Figures

List of Tables

Acronyms

AC3	Audio Compression - 3
ADC	Analog to Digital converter
AJAX	Asynchronous JavaScript and XML
ARPANET	Advanced Research Projects Agency Network
AVC	Advanced Video Coding
CLR	Common Language Runtime Environment
CSA	Cloud Security Alliance
DAC	Digital to Analog converter
DCT	Discrete Cosine Transform
DPCM	Discrete Pulse Code Modulation
FLAC	Free Lossless Audio Codec
HDTV	High Definition TV
IaaS	Infrastructure as a Service
IBM	International Business Machines
IDE	Integrated Development Environment
IT	Information technology
JIT	Just in Time
JSON	JavaScript Object Notation
LZ	Lempel-Ziv
LZ77	Lempel-Ziv Coding 1977
LZ78	Lempel-Ziv Coding 1978
LZMA	Lempel-Ziv-Markov
LZW	Lempel-Ziv-Welch
MIT	Massachusetts Institute of Technology
MPEG	Moving Picture Experts Group
MPEG3	Moving Picture Experts Group Layer 3
MPEG4	Moving Picture Experts Group Layer 4
MSIL	Microsoft Intermediate Language
NIST	National Institute of Standards and Tecnology
OAUTH	Open Authorization
PaaS	Platform as a Service
PAYG	Pay as You Go
PCI	Payement Card Industry
PGP	Pretty Good Privacy
REST	Representational State Transfer
RLE	Run-Length Encoding
SaaS	Software as a Service
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
URI	Universal Resource Identifier
VLC	Variable Length Coding
VOD	Video on Demand
W3C	World Wide Web Consortium
WMA	Windows Media Audio
XML	Extensible Markup Language
YAML	Yet Another Markup Language

Chapter 1

Introduction

1.1 Focus and Scope

Cloud computing is seen as the last revolution in information technology, providing cheaper services and agility and productivity of organizations. But, despite all the advantages there is still much discussion around the term Cloud Computing: what feature displays? What limitations in terms of storage, compression and security?

Data management in cloud computing and data storage is widely used and is available from different suppliers and different solutions. Can be used as a data repository or as infrastructure to support and implement applications [1].

Cloud computing presents interesting features for all users with advantages at the level of storage cost, scalability, elasticity face to the constant needs of availability and quick access in different computing devices. There are different types of services for cloud computing: Infrastructure-as-a-service (IaaS) where the cloud offers services of its infrastructure such as cpu, memory, and storage. Platform-as-a-service (PaaS) where it is available an execution environment for the user and Software-as-a-Service(SaaS) where the cloud provides a specific application accessible via browser [2].

The management and monitoring of data is an essential requirement for organizations, since it allows the reduction of the risk of internal breaches and ensures the responsibility of administrators. Each type of cloud service provides security requirements for the organization and access to data and systems.

Organizations that use cloud technology monetize their services extracting the greatest benefits of this new technology base. These companies have embraced the cloud as a strategy that provides everything you need to create, operate, manage and allocate in the cloud with efficiency, speed and reliability.The system of cloud computing provides a computational power available to the needs of the user thanks to a dynamic scalability [3]. However even considering the characteristics and advantages presented to maintain safe and reliable data presents itself as very problematic for users.

The purpose of this thesis aims to design, implement and evaluate a system that allows access to management, kept in cloud storage. The proposed application is intended to be used as intermediate system, aggregating a set of components and services between the user and various public clouds of data storage, made available by internet providers and cloud service providers.

1.2 Problem Statement and Objectives

The Cloud Computing still motivates much discussion. The essence of its functioning, its boundaries, the development of new applications, becoming increasingly agile and collaborative, inspiring subjects for research.

As we enter the new century, it appears that the ability of data centers is limited and runs out. The economy influences the trend of technological development and the solution is the adoption of grid services and/or utility computing as well as the use of virtualization to maximize the available resources. As services and applications become more distributed, paradigms like Service-Oriented Architecture emerge in response to integration and service orchestration, and the organization and technologies used in datacenters evolve. Today the data centers that support environments and platforms for Cloud Computing components are designed to utilize more economical, safe, quick and easy replacement. Given these factors, software developers have come, to design applications that can address the needs of users, in a safe, efficient and cost-effective way to implement and/or maintain an infrastructure of Cloud Computing.

Cloud storage services have grown and diversified significantly this development eventually promote the emergence of contract services and with the feature of allowing users to choose to acquire the one that is more suitable for him/her. Services are provided in a common environment by centralized cloud storage facilities. As data volumes processed by large-scale distributed dataintensive applications grow at high-speed, an increasing input/output pressure is put on the underlying storage service, which is responsible for data management. With the emergence of cloud computing, data intensive applications become attractive for a wide public that does not have the resources to maintain expensive large scale distributed infrastructures to run such applications [4].

The main objective of this dissertation is to propose and validate a solution to the problem of secure and efficient transmission and storage of multimedia content in public cloud environments using joint compression and encription. To answer this question, we investigated concepts of cloud computing, compression, transmission and storage, and security in the application, and proposed to build an application that allows the transmission and secure and efficient storage of multimedia content in two public cloud environments, Dropbox and MeoCloud, using compression and encryption together or one of them.

Complementing the overall goal explained above, we also intend to compress multimedia data using lossless algorithms to transmit and store quickly and efficiently. It allows reliable use of public cloud environment without constraints in terms of space, ensuring security in public cloud environments.

1.3 Adopted Approach for Solving the Problem

The use of cloud environments is revolutionizing the way we design, implement and manage the entire infrastructure of information technology and has created high expectations due to the advantages that this paradigm holds. However, organizations have questioned the feasibility of safe transition and questioned the data confidentiality. Therefore, according to the purpose of this dissertation, in order to build an application that allows secure and efficient transfer and storage of multimedia content in public cloud environments using joint Compression and Encryption. In this sense, this dissertation proposed to perform a literature review on the topic of cloud computing, secure storage and transmission of multimedia content, using lossless compression, in public cloud environments, and implement a system by building a platform that manages the data in public cloud environments (DropBox and MeoCloud).

To develop the application we used the programming software Microsoft Visual Studio 2010, for the implementation of the proposed application we chose to use the programming language C#. For the compression process it was used the LZMA algorithm, regarding the safety

of files we resorted to encryption using the AES algorithm.

1.4 Dissertation Overview

Followed by this introductory chapter, it is presented the state of the art of this dissertation that is organized into two chapters, chapter 2 - Background on Cloud Storage and chapter 3 - Data compression. Chapter 4 - Architecture, Prototype and Validation of the Framework for Secure and Efficient Storage of Multimedia Content in Public Cloud Environments present the validation and setting of the framework.

Chapter 2 was dedicated to cloud storage, providing an overview on the historical perspective and the evolution of cloud computing paradigm and described its architecture, models and cloud types, as well as the advantages and barriers for its adoption and security issues on cloud storage. In Chapter 3 devoted to data compression, it was performed a study on the compression methods with and without loss for the treatment of data, images, video and sound, with the specifics of each. In the second part this work, we highlight how we built the application, presented the features, advantages, and secure data. Finally, we reflect on the results, according to the theoretical framework made in the first part and platform development.

Chapter 2

Background on Cloud Storage

2.1 Introduction

At the begginning of this century, cloud computing has emerged in the computer science and engineering as a new paradigm in information technology, providing good services, increasing agility and productivity. Cloud computing refers to a computing model that moves all data to large storage centers, distributed in the form of Internet-based services, accelerating the availability of new products and services to market and increasing efficiency and resource utilization.

This chapter addresses the cloud computing paradigm, the historical context, the different models and architectures. Specifically for cloud computing are further highlighted the benefits that their use can provide to the users and barriers that may inhibit its adoption. It also explored the issue of security and the risks inherent in cloud storage. Finally, it is presented the related work to this topic and appropriate conclusions drawn.

2.2 Brief Overview on the History of Storage Systems

Manking always looked for ways to save his data. The evolution was gradually made from puched cards till cloud storage, from the year 1900 to 1950, punched cards, punched card, punch card, IBM card, or Hollerith card, were the primary means of data entry, data storage and processing in institutional computing, by IBM. These cards were the major precursors of memory used in computers [5].

Punched cards were progressively replaced by magnetic tape, in 1951. The idea of using magnetic tape to record information was presented by Oberlin Smith [6] [7]. The UNISERVO was the main device input/output on the UNIVAC I computer to use a tape drive, and in 1956 was used for the first time, a magnetic disk drive with a head of mobile reading. It was a commercial computer developed by IBM called Random Access Method of Accounting and Control [8]. In 1972 appears the cassette or compact cassette, a standard magnetic tape for recording audio, invention of the Dutch company Philips [5]. Initially the expectation was to put data on this support, but its price has led to it being put aside in favor of the diskette.

Goda [5] also states that in the seventies comes the floppy-disk, a disk for data storage. It is considered the first data storage device sold in mass for mass consumption.

In 1980, IBM 3380 Direct Access Storage Device is introduced. Used a new read head and had a capacity of 2.52 GB, at a rate of 3 MB per second of data transfer. This was the first storage device to achieve Gigabyte.

In 1980 also appears ST-506 that was the first hard drive on 5 "1/4. presented by Seagate Technology [9]. This was the first drive 5.25 "and was the one who gave birth to what we know today's hard drives.

In 1990 emerges the Compact Disc mainly used to record music or data. This support has marked the history by its widespread use currently and still manages to retain its presence

both in the music recording and data, and in 1993 the MiniDisc (MD) that stores information and usually privileges the audio. In 1994 appears the Zip Drive a removable disk system of medium-capacity , introduced by Iomega. Resulted in a disc with performance much faster than a floppy drive. [10].

In 1999, Microdrive launches a miniature hard drive of an inch designed to fit in a CompactFlash Type II slot. This device is in history for being prior to the flash memory as we know (cheap and with great offer) and present in the "boom" that was the appearance of the first Apple iPods. Apple has used this technology in the classic iPods. In 2000, the SD cards are commonly used in cameras, smartphones and GPS, to provide or increase the memory of these devices. There are many versions but the best known is undoubtedly the micro-SD memory card that works on most smartphones. The Memory Cards Secure Digital Card is an evolution of technology MultiMediaCard [5]. In 2008 appears the solid-state drive, a device with no moving parts to non-volatile storage of digital data. Marked the history by allowing impressive boot times and safe storage. Memories are fast, consume less energy and are the logical evolution of physical accommodation. They can also be combined with mechanical drives, having therefore a hybrid unit, such as Apple and fusion drive [11].

Nowadays we have the concept of cloud computing that refers to the use of memory, storage capacities and calculation of computers and servers shared and interconnected via the Internet, following the principle of grid computing.

2.3 Historical Context of Cloud Computing

Several researchers have sought to define the term cloud computing. Table 1 presents a summary of definitions proposed by known organizations and institutions [12].

Table 2.1: Definition to Cloud Computing. Adapted from [12].

Organization	Definition
The 451 Group	TI offered as a service, available through independent resources of physical location.
Wikipedia	A style of computing in which resources are dynamically scalable, usually virtualized, are provided as services over the Internet.
International Business Machines, IBM	Platform that provides dynamically, configures, reconfigures and releases servers according to needs, and employs large data centers and powerful servers in the hosting of applications and services for being used via Internet.
University of California, Berkeleu	Illusion of infinite computing resources available on demand, eliminating commitments, more effective for users of the cloud and enabling payment for the use of these resources according to the needs of short-term.

The term Cloud was first used as a metaphor for the Internet, "the Cloud of intermediate networks" [13] a thirty page report published in 1997 by the Massachusetts Institute of Technology (MIT). The historical background of Cloud Storage is systematized in Figure 2.1.

The term Cloud Computing appears much later when the CEO of Google, Eric Schmidt, at a conference in 2006 on "Strategies Search Engines", invokes it, indicating that Google will call its new business model Cloud Computing [14]. This would allow ubiquitous access to data and computing which will be located in a cloud of various servers, located in a remote location. In the same year Amazon announced one of the pioneers and most important services in Cloud Computing to the present day: the Elastic Compute Cloud (EC2), as part of Amazon Web Services [1]. However, this data organization has its origins in the sixtiess with John McCarthy, a researcher in computer science, which proposed that computing was organized as a public utility, in which a service agency charged a fee for its use [15] together with Joseph Carl Robnett Licklider, one of the developers of the ARPANET, which introduced the idea of an intergalactic computer network. Their vision was that everyone should be connected to each other by accessing programs and data from any site, anywhere [16] [15].

In the seventies, computer scientist John McCarthy (creator of the term Artificial Intelligence and List Processing Language) proposes the concept of cloud, which would be as available as public utility services [17]. In 1999, the company's of enterprise applications Customer Relationship Management, with the address salesforce.com gave rise to the first services available via the Internet. This initiative introduced a new idea in the market, the provision of enterprise applications with Service Oriented architectures via the Internet [18].

The next step in 2002 was the launch of Amazon Web Services. From the Amazon Mechanical Turk, Amazon provided services at the level of storage, computing and human intelligence. Four years later, Amazon has launched a Web commercial service, ElasticComputeCloud. This service was intended to provide computer rental to small business users to implement their software applications [19]. In 2006 came the Simple Storage Service, a web service that Amazon offers, a scalable storage infrastructure with reduced costs. Their form of compensation was based on a pay as you go, in other words, the user pays for the resources used during the contracted period. This model became the payment model inherent to Cloud Computing [20].

After three years, Google company started its application offerings (Google Apps). Accessed by browsers, applications provided reliable services that are easy to consume by its users. With them, and with the gradual implementation of broadband worldwide, it has become the first step for interconnection and service discovery.

1960 ARPANET	**1999 SalesForce. com**	**2002 Amazon Computing and Storage Web Services**	**2006 EC2 and S3 Scalable Services PAYG**	**2009 Microsoft IBM and Google Service Discovery**

Figure 2.1: Evolution of Cloud Computing. Adapted from [21].

The evolution of the Cloud Computing paradigm can be systematized in the evolutionary diagram shown in Figure 2.2, which refers the paradigms that have contributed to the appearance of Cloud Computing [21].

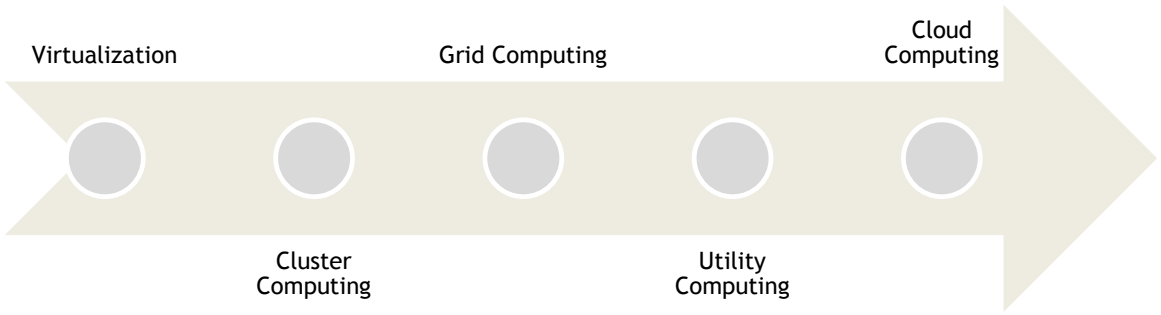

Figure 2.2: Schematic of the evolution of cloud computing paradigm. Adapted from: [21].

Virtualization - Virtualization is a concept that means to introduce an additional layer between the physical system and applications which translates concurrent accesses to the physical system providing exclusive access to virtual resources while abstracts the physical resources [22]. Virtualization is a key technology to support cloud computing, as well as for grid computing. Almost all providers of Cloud Computing abstract the hardware layer using some sort of virtualization [23]. With the use of this controller it is possible that multiple operating systems on the form of virtual images be executed concurrently in the same environment. At the level of virtualization software platforms we can cite many examples such as Windows, Unix or Java which are among the most important ones.

Cluster Computing - The Cluster Computing was designed to allow the increased use, scalability and availability of computing resources individualized. Physically, a cluster is composed of interconnected computers that originate from a single integrated computing resource. This model is a parallel and a distributed system, whose resources are located in the administrative domain and managed by an organization [24].

Grid Computing - The Grid computing paradigm has emerged with the goal of integrating heterogeneous and geographically distributed resources. The Grid Computing infrastructure is supported by an aggregate of clusters, composed of heterogeneous and geographically distributed resources [25].

Utility Computing - says that utility computing is the delivery of infrastructure, applications and business processes in a safe, shareable, scalable, standard-based computing environment, from the Internet by paying a monthly fee. Customers subscribe to these services and pay for them, in a simple and similar to what is done with water or electricity services [26].

2.4 Brief Overview on Cloud Computing

2.4.1 Reference Model for Cloud Computing

The National Institute of Standards and Technology(NIST) proposes a reference model for cloud computing [27], which considers the existence of three distinct dimensions: Essential Characteristics, Service Models and Deployment Models. These three dimensions are described below:

a) Essential Characteristics [27]:

- On-demand self-service - A consumer can unilaterally provide computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

- Broad network access - Capabilities are available over the network and accessed through standard mechanisms that promote the use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

- Resource pooling - The providers computing resources are combined to serve multiple consumers using a multi-client model, with different physical and virtual resources, dynamically assigned and reassigned according to consumer demand. There is a sense of location independence since, the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify the location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

- Rapid elasticity - Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

- Measured service - Cloud systems automatically control and optimize resource use, by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g. storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the used service.

b) Service Models [27]:

- Software as a Service (SaaS) - The capability provided to the consumer, ability to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

- Platform as a Service (PaaS) - The capability provided to the consumer is to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

- Infrastructure as a Service (IaaS) - The capability provided to the consumer is the

provision of processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

c) Deployment Models [27]:

- Private cloud - The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

- Community cloud - The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

- Public cloud - The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

- Hybrid cloud - The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

The Cloud Security Alliance complements the NIST reference model, providing a graphical representation of the same and, in addition, supports the inclusion of more an essential feature in the model, namely the multi-tenancy, which she considers essential for applications operated in clouds. Multi-tenancy is the ability of modern applications have to, process data regardless of different users from a single instance running.

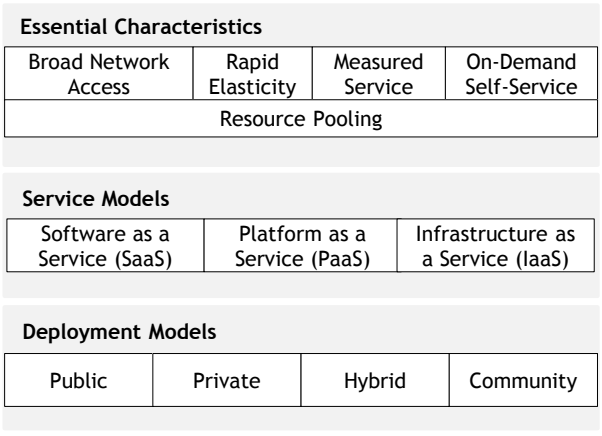

Figure 2.3: Adapted from NIST Visual Model of Cloud Computing Definition [28].

Some authors accepts the three dimensions proposed by NIST but also considers the existence of an additional type of service offering in the model, that proposes the database as a service (DaaS) [14].

The technology of cloud computing is a reality and given the evolution of applications towards portability and mobility, the concepts of notebook or desktop are no longer enough. The world is looking for connectivity, integration, speed and agility, items found in cloud computing. At the same time creates a large volume of information, people want devices ever smaller, lighter and more agile and the answer for this demand that is the cloud computing. With this technology, companies and users have instant access to the files anytime, anywhere, through various types of devices, such as desktops, smartphones, tablets and netbooks connected to the Internet with maximum flexibility. Cloud computing is a reality increasingly strong in the market. It opens new possibilities in the business world, the cloud can be part of a new model of communication, the ability to offer outsourced services, as well as a strategic solution for the expansion of new markets [29].

Jensen et al. [30], use the taxonomy of scale service offerings NIST to propose a simpler architecture for cloud computing, based on that scale and vision of the technologies used to enable access to service offers. Access technologies considered in this case, include web browsers and web services and their interaction with the offers is given as follows:

- Web browsers fully support SaaS offerings and a portion of PaaS;
- Web services are components of IaaS offerings and part of PaaS.

Jericho Forum [31], a study group linked to The Open Group, proposes a model structured around four dimensions, the cube of the cloud. The considered dimensions, relate to:

- Location of the data: internal or external to the organization to which they belong;
- Nature of computational resources from the cloud: proprietary or open;
- Architecture of the environment: limited to a perimeter or unlimited;
- Ownership of resources allocated to the cloud or to a third-party organization that uses the cloud.

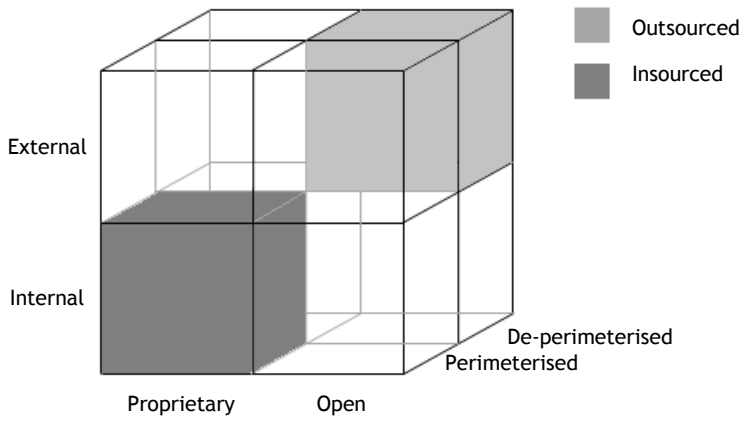

Figure 2.4: Cloud Computing Cube. Adapted from [31].

In short the cloud computing covers the said spaces called clouds which are environments that have resources (hardware, development platforms and/or services) virtually accessible and easy to use. The data storage in cloud computing can be accessed at any time without installing software. These features can be dynamically reconfigured due to virtualization in order to adjust to a given variable, thereby permitting optimum use of resources. This system provides access to information, files and programs into a single system, regardless of the platform, with a simple computer, compatible with the resources available on the Internet.It is created therefore a conceptual layer - a cloud - that hides all the infrastructure and resources, but that presents a standard interface that provides a multitude of services. Once the user connects to the Internet, he/she has all the resources at his disposal, suggesting a power and infinite capacity [32].

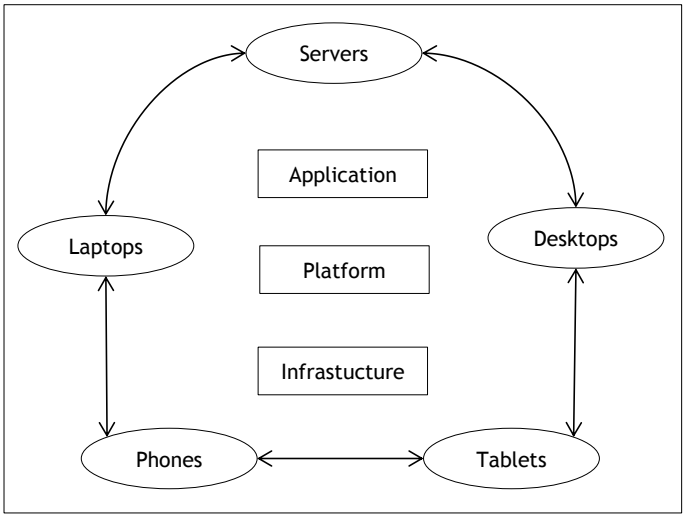

Figure 2.5: Cloud Computing

For the computing infrastructure to be located on the network, applications and data from computers are moved to large data processing centers, data centers. Systems hardware and software in the present data centers come from applications in the form of services on the Internet [33]. These environments are usually explored through a model pay-per-use. [24].

In technological terms, Cloud Computing represents the unification of the capacity of all resources on the Internet, even suggesting a multitude of resources in terms of power and capability. The cloud computing has significant power to provide a degree of local computing and caching support, which tends to grow in terms of users and volume [32].

2.4.2 Architecture of Cloud Computing

- **Actors** - Cloud computing consists of three main actors [24], service providers who develop and leave the services accessible to users, service users, also known as clouds users and providers of infrastructure.

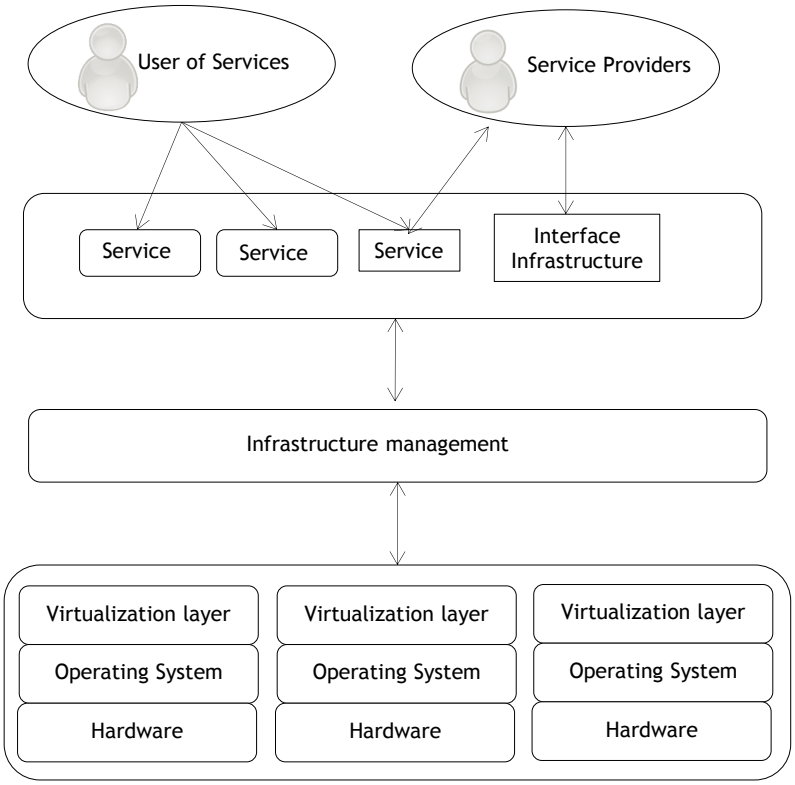

Figure 2.6: Architecture of Cloud Computing. Adapted from [24].

- **Layers** - The architecture of cloud computing can be divided into three abstract layers [33]. The infrastructure layer is the lowest layer. It is through it that the providers of infrastructure (servers, storage systems, such as data centers, and routers) provide network services and cloud storage. The platform layer has a higher abstraction and provides services so that applications can be developed, tested, implemented and maintained in a cloud environment for service providers and application layer is the highest level of abstraction, it is the one that offers various applications as services to users.

- **Scenarios** Cloud computing resources are distributed in the form of services. These services may be available in any particular abstract layers:

- Application as a service (Software as a Service - SaaS) - The concept of SaaS is the culmination of the cloud opportunity for business customers. Applications are made available as a service on PAYG payment model. Consumers only pay for what used and/or the duration of the use of resources. This mode provides the reduction of complexity inherent to the configuration and maintenance of the infrastructure (e.g operating system, hardware), because these activities are the responsibility of the supplier. The user only needs to worry about access to the service, obtained through a device with network connection and an interface that will facilitate access.

- Platform as a Service (Platform as a Service - PaaS) - The concept of platform as a service, PaaS, consists of a layer of programmable software for development and installation (deploy) of high-level services [34]. The goal of PaaS is to facilitate the development of applica-

tions, intended for cloud users, through a platform that streamlines the process of integration, regardless of geographic location, worry-free management and resource allocation. From the platform resources are made available, (e.g operating systems), for users to provide their SaaS applications [35].

- Infrastructure as a service (Infrastructure as a Service - IaaS) - IaaS is the level of service innermost in Cloud Computing and that supports others, PaaS and SaaS. Enables to interested parties on the use of resources, servers, network and storage devices, and other computing resources available. The infrastructure consists of computing resources allocated to support scalable environment, with support for layers and respective middleware applications.

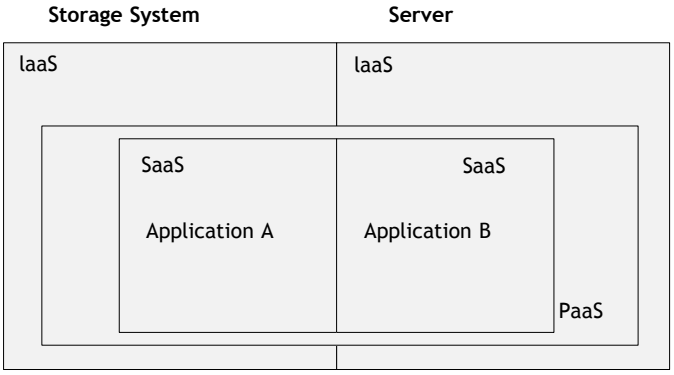

Figure 2.7: Cloud Computing Scenarios

2.5 Advantage and Barriers for the Adoption of Cloud Computing

2.5.1 Advantages in the Use of Cloud Storage

The storage in clouds brings great advantages over traditional storage. Store in a cloud, allows access from any location with Internet access and avoids the necessity of maintaining a storage infrastructure in the organization. Armbrust et al. [36] highlights three main aspects in cloud computing:

- The illusion of unlimited availability of resources, the concept of the cloud suggests that the user has in his hands the entire Internet and its services;

- The elimination of a commitment on the part of users. A company can start using few hardware resources and, as necessary, could increase the amount of resources used;

- The ability to pay for the use of resources as they are used, allowing the release of funds if not used, avoiding unnecessary consumption.

Cloud computing offers big benefits to users:

- Reduced cost of initial investment,
- Easy to scale services according to the clients claim,
- Innovation in services that were previously impossible.

2.5.2 Barriers for the Adoption of Cloud Services

Despite the significant benefits, many organizations collective and individual, are hesitant to adopt the Cloud Computing. They cite concerns about reliability, putting security and compliance as the first barrier. When Gartner evaluated the barriers to cloud adoption in 2009, the most important were related with confidence. The concerns range from protection of confidential business information to comply with legal requirements and the protection of personal data. The authors Westerman and Hunter [37] argue that the risk factors presented by them form a hierarchy, which describe as Pyramid IT risk. Westerman and Hunter [37] indicate that the causes of these risks are associated much more to governance problems than technical problems.

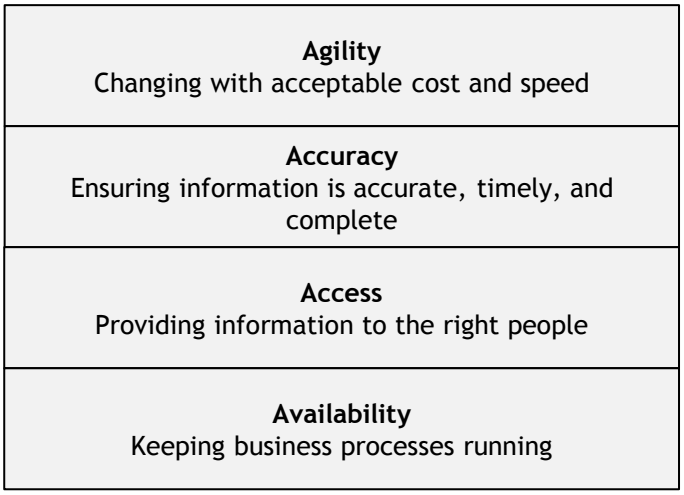

Figure 2.8: Pyramid IT risk. Adapted from [37].

Westerman and Hunter [37] created the Pyramid of IT risk (known as framework 4A) shown above. The 4A framework is a mechanism to translate IT risk in business terms. This framework defines IT risk as the possibility that some unforeseen event involving the IT threaten any one of four interrelated objectives of the company: availability, access, accuracy and agility. Any IT risk must be understood in terms of its potential to affect all business objectives mediated by IT. The pyramid shape induces the fact that each element of a layer influences not only the risk that layer, but also the risk of the layers above.

Security is a priority concern for many consumers of cloud computing and has a share of influence on purchasing decisions. This consumer behavior should be seen by providers of cloud services as an important driver for the improvement of their security practices [38].

Several authors have addressed the issue of the risks inherent to cloud computing:

- Kim [39] cites the following risks that may affect the service consumers of cloud computing: The availability, privacy, support, entrapment, and interoperability and compliance.

- Smith [12] also offers a list of risks to be considered: privacy, legislation, performance, errors, trapping, finite capacity,

- Armbrust et al. [36], also engage with this theme, identifying barriers for the adoption and the growing use of cloud computing: the availability of services, the imprisonment of data, confidentiality and the audit data, the data transfer bottlenecks, unpredictability of performance, scalability, storage, errors in distributed systems for high-scale, rapid escalation, the bad reputation of the software licensing.

The CSA states that similar to what occurs with every other aspect of business, constitutes a good practice to adopt an approach based on risk assessment and security, when considering whether to hire services of cloud computing.

2.6 Cloud Storage Services

2.6.1 Models of Infrastructure Cloud Services

Public clouds are available for any individual, from Internet access, the user can access it and start using almost immediately, private clouds are typically located inside a private infrastructure under the control of an organization. Each entity shall perform the analysis and is users responsibility to choose between models, public, private or hybrid, taking into account a range of factors [40]: The initial investment, the volume of information, data longevity, performance, access patterns and the location, security and isolation of information, confidentiality and destruction of information, service level agreements and own technical resources. Should take into account all these considerations, opting for the most suitable model (public or private). In some cases it can be used the hybrid model. To meet temporary needs, an application may be located in a public environment, avoiding the need to anticipate investment in infrastructural component of the organization, to resolve a temporary issue.

Public clouds are run by great services and applications of different clients executed simultaneously on shared servers. They also share the same storage device information and network structure.

Private clouds are available for use for a single user, providing a total control over information, security and quality of service. The organization has the infrastructure and control how the applications are installed and run. Private clouds can be implemented in data centers or a private provider of cloud computing services. This model of private cloud hosting service provider, will be responsible for installing, configuring and operating the entire infrastructure that supports it. This provides the organization that implements it, secured the level of control of information and resources but also knowhow requirements related to the implementation, integration and maintenance.

Hybrid clouds help in provisioning external resources promoting the scalability of the infrastructure. This is a common situation in the use of storage services to support Web 2.0 applications. The hybrid model can also be used for planned scalability issues in which are well known the load fluctuations of the application. According to this model, a public cloud would play the role of processing periodic tasks which represent a volume of considerable burden, thus freeing resources of the private infrastructure. The architecture of the solution would be dispersed by the two environments. Another example relates to the use of one being the backup of the other environment that would come into operation if a fault was detected in the primary environment [41]. However, a hybrid model introduces complexity in determining how applications will be distributed and/or components for the two infrastructures.

2.6.2 Provision of Cloud Storage Services

It is available on the web a considerable number of storage providers in clouds and storage space offered to regular clients grow. There are suppliers of storage clouds that charge a fixed amount for a share of space and bandwidth of input and output data, while others use a model pay-per-use and charge varying amounts depending on the space occupied and the

bandwidth used by the client. The model collection of pay-per-use incorporates the concept of elasticity of resources: pays only when using the service and can grow arbitrarily to accommodate sporadic demands.

There are several storage services in clouds (Cloud Storaging) such as:

- **Clouds Pay-per-use;**

-Amazon Simple Storage Service, with location in the United States (3 centers), Ireland and Singapore, http://aws.amazon.com/s3/

-Microsoft Windows Azure Platform, USA (Chicago) and Ireland (Dublin), http://www.windowsazure.com/en-us/

-RackSpace, with location in the U.S. (Texas (3 centers), Virginia (2 centers) and Chicago), UK (2 centers) and Hong Kong, http://www.rackspace.com/pt/.

- **Clouds of fixed cost;**
  DivShare(http://www.divshare.com/),
  DocStoc(http://www.docstoc.com/),
  Box.net(https://www.box.com/),
  FilesAnywhere(https://www.filesanywhere.com/) and others.

The price of cloud storage environment have been coming lower due to the multiplicity of authorities in this area and sometimes the offer of free services is only limited in terms of space.

2.6.3 Some Existing Applications for Cloud Storage

The use of information technologies by individual users and groups has faced a rapid paradigm shift. Recently we have seen an increasing use of cloud services that support a larger number of users and parallel applications are available that perform migration services in the cloud such as among others Cloud Migrator (http://www.cloudberrylab.com/cloud-migrator.aspx) and Egnyte (http://www.egnyte.com/file-server/online-file-server-features.html). These applications provide to the user various services, Cloud Migrator enables the scheduling of services to copy data from cloud to cloud or from account to account, Egnyte ensures the secure access to files, shares, and secure storage.

Specifically in this work, and as previously stated, the application built, was designed, ensuring a range of services to users across multiple storage clouds. Thus the application ensures the maintenance, lossless compression, ensuring the properties of security, privacy and reliability as from the encryption. It is considered that the services provided by this application and available to the user, are advantageous but also innovative because usually the various applications available in the market only offer migration services.

2.7 Security Issues in Cloud Storage

With the growing evolution of the Web, Internet gave access to numerous services and information. This breakthrough stimulated the proliferation of information which is in part responsible for the formation and development of society. The information is assuming more and more, a strategic position for organizations, being its main asset. In this sense, the control of access to information is a critical requirement in systems of organizations; seen that currently

the vast majority of an organization's information is stored and is exchanged between its various systems.

Cloud computing is inserted in the informatics as the last revolution in information technology, providing, increasing their agility and productivity. Despite the significant benefits, many organizations/institutions are hesitant to adopt cloud computing, mentioning concerns, putting security and compliance as barriers to their use. Concerns go from protection of confidential information to compliance with the legal requirements and the protection of personal data. The choice of data storage services maintained by third parties is only realistic if are guaranteed security and privacy properties, well as reliability and continuous availability. For this end, these properties must be preserved under control and independent audit by the users. Only this way, users can adopt these services as reliable systems.

It is therefore necessary that the properties of security and reliability, such as: authenticity, privacy, integrity, access control, intrusion or fault tolerance, recovery, and availability of data, kept in cloud storage providers on the Internet, be audited and controlled by the users themselves.

In todays clouds the primary security mechanism is virtualization. Virtualization is a powerful defense, and protects against most attempts by users to launch attacks against other users. However, not all resources are virtualized and not all virtualization environments are bug-free. Virtualization software is known to have some flaws that allow which allows to visualize portions of virtualized code. That is, incorrect network virtualization may allow user to have access to sensitive code portions of the providers infrastructure, or to other users resources. These challenges, though, are similar to those involved in managing large non-cloud data centers, where different applications need to be protected from one another. Large Internet services will need to ensure that a security problem doesnt compromise everything. One last security concern is protecting the cloud user against the provider. The provider will by definition control the "bottom layer" of the software stack, which effectively circumvents most known security techniques. Absent radical improvements in security technology, we expect that users will use contracts and courts, rather than clever security engineering, to guard against provider malfeasance [36].

2.7.1 Reliable Storage and Management of Data

To provide a reliable storage and management of data, certain aspects should be taken into account, according to Albuquerque and Ribeiro [42], there are three basic requirements to ensure information security:

- **Confidentiality**: which ensures that information can only be accessed by authorized persons. Therefore, there should be no disclosure of undisclosed information.
- **Integrity**: The information must be retrieved in its original form (when it was stored).
- **Availability**: The system must always be available, even in the event of failure. Many vendors are unable to ensure the availability of the system at all the time. There is record of situations where systems were unavailable for a certain period, damaging several businesses companies. It appears that the best approach is to rely data to a set of clouds, instead of just one.

Other authors introduce other requirements which they say ensures the safety of the cloud:

- **The Reliability** A system with high reliability is seen as being reliable and even in the presence of failures keeps its correct operation. The reliability can be ensured using Byzantine fault tolerance algorithms [43], based on data replication. The distribution of the data replicated by a set of public cloud offers high reliability, beyond this is also the one that is provided individually for each cloud.
- **The Privacy** The stored data must previously be encrypted, to preserve its confidentiality. It is necessary to ensure that sensitive data never be exposed, even in the event of an attacker have access to them.
- **The supplier Independence** (vendor lock-in) The auditability and control in the user's side guarantee independence conditions from the suppliers, autonomous control, data maintenance and management. This requires resiliency solutions auditable by the users that withstand eventual service breaks, intrusion at the level of infrastructure or applications providers, or possible incorrect operation, malicious or illicit by personnel from operative teams of providers. It is also important that these solutions prevent interlocking shapes or avoid illegal business practices that can be exploited by suppliers, in relation to the users data.
- **The Scalability:** Scalability is also a very important property. A scalable system and with a fast response time to changes, ensures that resources are always properly allocated. This ensures that these are not being used unnecessarily or that are not missing resources. The authors Rezende and Abreu [44] and Sêmola [45] defend that for information be deemed safe, shall also respect the following criteria:
  - **Authenticity:** This requirement provides assurance of the origin of the data and the identity of the person or system. The authentication service must indicate whether the message is really coming from the source indicated in its content.
  - **Non-repudiation:** The user can not deny (in the sense of saying that it was not done) that performed an operation that modified or created information in the system. It is not possible deny the sending or the reception of an information.
  - **Audit:** Inspection of the various steps of a business or process, and consequently the state of information, increasing its credibility.

Filho and Neto [46] indicates that the concepts identified above aim to provide information systems against various types of threats such as:

- Disclosure of information - in espionage cases;
- Fraud - non-recognition of the origin, modification of information or even espionage case;
- Interruption - modification of information;
- Usurpation - modification of information, denial of service or espionage.

Threats can be of different natures and are, generally, classified as passive, active, malicious, not malicious. To deal with these threats, becomes necessary to define security policies and mechanisms, supporting prevention to prevent attackers from violating security mechanisms, detection which is the ability to detect intrusion in security mechanisms and recovery which consists in the mechanism that stops the threat, evaluate and repair damage, besides keeping the system operational in the event of invasion of the system.

Security is a major concern for companies interested in implementing a cloud computing strategy. The vice president of Gartner, John Pescatore [47] state that there are ways to mitigate the fears, building security measures specifically designed to protect applications data or work volumes based on cloud computing. Gives as an example the certification Payment Card Industry that demands encryption of all customer data of their credit card stored electronically. Pescatore [47] states that the focus of security in cloud computing should focus on protection processes of cloud computing. Should be developed security policies on cloud computing, then, make sure they are deployed all over the cloud system and remain in compliance with them.

2.7.2 Threats Classification

According to the study from Estrela [48], threats can be classified according to four categories:

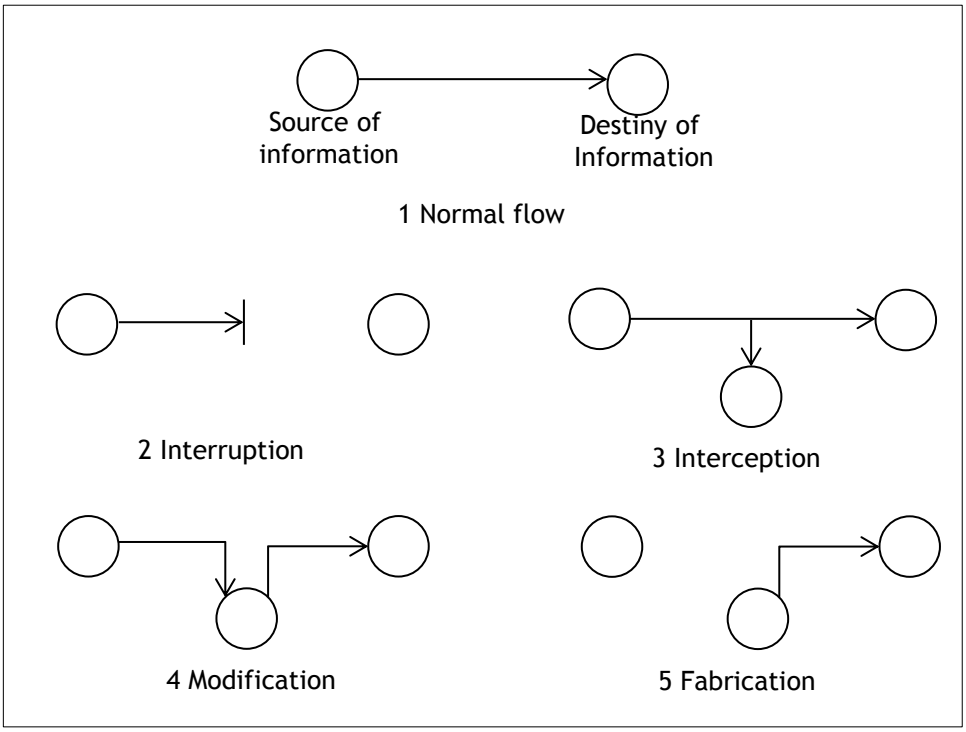

Figure 2.9: Threats Classification. Adapted from [48].

- **Interruption:** A feature of the system is destroyed or becomes unavailable. This is a threat to the availability of information.
- **Interception:** An unauthorized party gains access to a resource. This is a threat to the confidentiality of information.
- **Modification:** An unauthorized party not only gains access but also interferes with a resource, becoming a threat to the integrity of the information.
- **Production/Manufacturing:** An unauthorized party inserts counterfeit objects on the network. It is a threat to the authenticity of the information.

If a threat materializes becomes an attack or an Incident of Information Security, defined in ISO/IEC 17799:2005 as one or more events that are likely to produce significant damage.

According Estrela [48], these attacks can be divided into two categories:

- **Passive Attacks:** Are attacks that consist in listening or monitoring transmissions. The goal of the attacker is to obtain information that is being transmitted. Passive attacks involve two types of action: obtaining message content and traffic analysis. Passive attacks are very difficult to detect since it does not involve any modification of data. However, it is possible to prevent them.
- **Active attacks:** These attacks involve some modification of the data stream or the creation of false data and can be subdivided into four types;
  - The masking, when an entity pretends to be another, allowing an entity with few privileges to obtain extra privileges, pretending to be an entity that has.
  - Repetition, which involves the capture passive of data and its subsequent retransmission.
  - The modification of messages, when a legitimate message is altered.
  - The denial of service prevents or inhibits the normal use of network resources, turning them off or saturating them with messages in order to degrade its performance.

2.7.3 Mechanisms to Control Threats

In Information Security exist some mechanisms to preserve the information, in order to ensure their availability, confidentiality, integrity and authenticity, These mechanisms are referred as control mechanisms of threats.

- **Access Control:** This mechanism lets you control which persons are authorized to enter certain place and records the day and time of access, controlling and deciding which permissions each user has. An access control system is composed of different peripheral devices of command and control, interfaced to a single control unit that allows, at different points, multiple accesses.
- **Intrusion Detection:** The intrusion detection systems alert administrators to the possible entry of intruders in the system. These systems attempt to recognize a behavior/action intrusive, through analysis of available information in a computer system or network.
- **Cryptography:** Cryptography is the art of coding that allows the reversible transformation of information in order to make it intelligible to third parties. This algorithm uses a certain secret key applied to a set of unencrypted data, to produce a sequence of encrypted data.

Cryptography is an ancient practice that aims to encode information so that only the sender and receiver are able to decrypt it. The emergence of the Internet and the ease that it provides in data transmission made such practice an essential resource to safeguard data safety. According to Estrela [48] is the ideal solution to the need to communicate over insecure connections without exposing the system. In Soares [49] encryption is defined as the study of the principles and techniques by which information can be transformed from their original form to another illegible, so that it can only be known by the recipient, making it difficult to be read by an unauthorized party. Thus, only the receiver of the message can read information with ease.

At present the codes are no longer used and have been replaced by the use of ciphers a

more practical and safer and better adapted to computers. Since the digital data is represented by bits, the encryption process is performed from algorithms that randomly shuffle the bits of the data using a particular key or key pair, depending on the chosen cryptographic system. Cryptography is widely used on the Web, as security to authenticate users to provide them access, in protecting transactions and communications networks.

However, although encryption is an important feature in the transmission of information via the internet, can not guarantee complete security, since there is always a way to decode the code.

Thus, the techniques are constantly being refined and complex and others are created. Today, among the most popular techniques is the concept of cryptographic keys, in which a set of bits based on a particular algorithm can encode and decode information. There are two types of ciphers, the symmetric and asymmetric. If the receiver of the message resolves to use a key incompatible with the issuer's key, the information will not be shared. There are still other concepts involved in cryptography, as the Hash function, used in digital signatures to ensure integrity and applications, such as digital certification.

The advancement of intrusion techniques and data interception forced the consequent evolution of cryptography, which adopted encodings of 256, 512 and up to 1024 bits. That means that are generated 21024 different combinations of keys for each message sent, of which only one is correct, known only by the sender and receiver.

Philip R. Zimmermman distinguished himself in the field of cryptography, developed in 1991, PGP (Pretty Good Privacy) is a software encryption and digital signatures for e-mail that uses asymmetric cryptography. Available free of charge, has become one of the most popular means of encryption. The software can also perform a second type encryption using a session key method which is a type of symmetric crypt.

Some open-source utilities that use encryption are:

- **OpenSSL** (http://www.openssl.org/): This tool is seen as the best cryptographic library SSL/TLS (Secure Socket Layer/Transport Layer Security) which are cryptographic protocols that provide security for communications over networks such as the Internet.

- **OpenSSH/SSH** (http://www.openssh.com/) is both a computer program and a network protocol that allows connection to another computer on the network. It has the same functionality as Telnet, with the advantage of the connection between client and server is encrypted.

- **OpenVPN** (http://openvpn.net/): Software of virtual reliable network that provides secure communication services, not only fulfilling the requirements of traditional VPN commercial, but also responding to the demands of the next wave of Web services VPN.

- **Truecrypt** (http://www.truecrypt.org): Is a software open-source of disk encryption for Windows, Linux and MacOS X. Users can encrypt entire file system (file and folder names, contents of every file), encrypting and decrypting it when is need it, without user intervention beyond inserting initially his password.

However, it is not always possible to ensure all the safety requirements, due to vulnerabilities that computer systems have. Even with these protective measures, many people believe that information stored on a remote storage system is vulnerable. There is always the possibility of somehow having access to information. There is also the possibility of the company's employees with access to the server can steal, change or destroy information. Companies in the

business of storage in clouds invest a lot of money on security measures to limit the possibility of theft or corruption of information. Furthermore, there is always the worry of putting critical data (and often confidential) in the hands of third parties who have access to the information contained in them. There is also the question of the reliability and availability of storage services. Store information on a remote system accessible via the Internet puts the organization vulnerable to all the problems of connectivity and temporary unavailability of the Internet.

According to Antone Gonsalves Computer World [50], The Massachusetts Institute of Technology, has developed a new security technique based on homomorphic encryption, that is not yet ready for the market, making it possible for a cloud server to process data without decipher it.

The new method involves the combination of homomorphic encryption with two other techniques, providing a functional encryption scheme. The research team recognizes that it takes a lot of computing power, but now that researchers know to be possible to process the data without the decrypt it, this problem can be addressed over time.

The homomorphic encryption makes possible to process the data, keeping the data encrypted from point to point. The new functional encryption scheme allows the server in the cloud running a single calculation specified on the outcome homomorphic encrypted form, without having to extract any other information. For this, the researchers used two other systems, called discontinued circuits and attribute-based encryption. Each has capabilities for functional encryption.

The new system starts with the homomorphic encryption and incorporates in the decryption algorithm a discontinuous circuit. The key to this circuit is in turn protected by the encryption by attributes for keeping the whole process ciphertext. Steve Pate, co-founder and CEO of Cipher HighCloud, explains that the new research is "encouraging". But it highlights a major challenge: "Computation required for homomorphic encryption far exceeds what exists in computing resources". Before the technique to function, there is a need for advances in hardware, where the cipher key management occurs within the processor or other hardware module. Once proven in the academic world, the technique has yet to be tested in the real world.

- **Authentication Systems:** According Serrão [51] authentication is the process for checking or testing whether a given identity is valid or not, requiring the user to supply additional information that must exactly match the professed identity. The key username/password is the most widely used system, however is not at all the safest, even being one of the least efficient, and easy to manipulate and decipher.

- **Digital Signature:** This mechanism is a set of encrypted data, associated with a document that ensures its integrity and authenticity. The use of digital signature proves that a message is from a particular sender, because it is a process that only the signer can perform. However, the receiver must be able to verify the signature made by the sender and the message can not be changed, otherwise the signature will not match over the document. The validity of a digital signature is verified if it is based on certificates issued by accredited certification entities.

- **Protection of Stored Data:** In this mechanism are used antivirus software that are able to detect and remove harmful programs or files. Concern for the protection of stored data causes them to develop some methods to control access by outsiders, such as encryption or digital signature.

- **Disaster Recovery:** Natural disasters (fires, floods, earthquakes, etc.) are designate disasters and are events that can cause large losses, however, with low probability of occurrence. However lead us to the need to implement emergency plans, to ensure the preservation of documents and physical safety of the employees of an organization.

To meet the needs of security, customers have a wide variety of options. Users should seek their providers are certified and for especially confidential information, there is offers security systems of third parties. Some service providers offer services encryption integrated into their storage offerings. There is a range of applications to who customers can purchase the services of encryption, protection Denial of service and access control measures specifically adapted to cloud computing.

2.7.4 Related Work on Secure Cloud Storage

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, several authors have carried out investigations in this area:

- Wang and Ren [52] refers that In order to facilitate rapid deployment of cloud data storage service and regain security assurances with outsourced data dependability, efficient methods that enable on-demand data correctness verification on behalf of cloud data owners have to be designed. Wang and Ren [52] propose that publicly auditable cloud data storage is able to help this nascent cloud economy become fully established. With public auditability, a trusted entity with expertise and capabilities data owners do not possess, can be delegated as an external audit party to assess the risk of outsourced data when needed. Such an auditing service not only helps save data owners computation resources but also provides a transparent yet cost-effective method for data owners to gain trust in the cloud.

- Wang et al. [53] say that Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This workstudies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing [53].

- Harnik, Pinkas and Shulman [54] refer that as the volume of data increases, so does the demand for online storage services, from simple backup services to cloud storage infrastructures. Remote backup services give users an online system for collecting, compressing, encrypting, and transferring data to a backup server provided by the hosting company. The term data deduplication refers to techniques that store only a single copy of redundant data,

and provide links to that copy instead of storing other actual copies of this data. As storage services transition from tape to disk, data deduplication has become a key component in the backup process. By storing and transmitting only a single copy of duplicate data, deduplication saves both disk space and network bandwidth. Deduplications effectiveness depends on such factors as the type of data, the retention period, and the number of users. However, theres an inherent risk in entrusting data to the storage cloud. In doing so, the data owner releases control over the data. Yet, a wide range of users and applications are more than willing to hand over their data storage tasks to cloud providers. They put their trust in the cloud providers integrity and in the security of its access control mechanisms. Setting these issues aside, we point out an additional threat: the privacy implications of cross-user deduplication. We demonstrate how deduplication in cloud storage services can serve as a side channel that reveals information about the contents of other users files. Deduplication can also serve as a covert channel through which malicious software can communicate with a command-and-control center, regardless of any firewall settings at the attacked machine. We analyze deduplications security issues and propose a simple mechanism that allows cross-user deduplication while reducing the risk of data leakage. Our mechanism states rules by which deduplication can be artificially turned off. This simple practice gives clients a guarantee that adding their data to the cloud has a limited effect on what an adversary might learn about this data. Thus, we can essentially assure clients of the privacy of their data [54].

- To solve this problem and further achieve a secure and dependable cloud storage service, Wang et al. [55] proposes a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server.

- Zhu et al. [56] propose a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. The audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, Zhu et al. [56] propose a method based on probabilistic query and periodic verification for improving the performance of audit services. The experimental results not only validate the effectiveness of our approaches, but also show that audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata.

- The storing data in a third partys cloud system causes serious concern over data confidentiality. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. Lin and Wen-Guey Tzeng [57] propose a threshold proxy re-encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted

messages as well as forwarding operations over encoded and encrypted messages. The method fully integrates encrypting, encoding, and forwarding.

- In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. Yang and Jia [58], design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, extends the auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model.

- For a large scale cloud cluster, providing data access which crosses through high performance computing servers, network and storage servers, and scheduling resources among multi-tenants are challenging jobs, especially with specific level of quality of service (QoS) requirements. Besides the criteria of current availability and reliability in service level agreement (SLA), throughput and latency are also critical QoS performance measurements to the tenants. Any QoS solution, unfortunately, cannot be achieved without introducing additional overhead. A good QoS design should always be cost effective. Yu et al. [59] propose a cost-effective QoS infrastructure over a distributed storage system, providing work-conserving differentiated QoS for multi-tenants cloud storage. The proposed QoS infrastructure combines global service differentiation with adaptive share allocation to support differentiated QoS and achieves both service isolation among users from the same class and service differentiation among users from different classes.

- Ming Li et al. [60], refers that cloud computing is envisioned as the next generation architecture of IT enterprises, providing convenient remote access to massively scalable data storage and application services. While this outsourced storage and computing paradigm can potentially bring great economical savings for data owners and users, its benefits may not be fully realized due to wide concerns of data owners that their private data may be involuntarily exposed or handled by cloud providers. Although end-to-end encryption techniques have been proposed as promising solutions for secure cloud data storage, a primary challenge toward building a full-fledged cloud data service remains: how to effectively support flexible data utilization services such as search over the data in a privacy-preserving manner. Ming Li et al. [60] identify the system requirements and challenges toward achieving privacy-assured searchable outsourced cloud data services, especially, how to design usable and practically efficient search schemes for encrypted cloud storage. We present a general methodology for this using searchable encryption techniques, which allows encrypted data to be searched by users without leaking information about the data itself and users queries. In particular, discuss three desirable functionalities of usable search operations: supporting result ranking, similarity search, and search over structured data. For each of them, describe approaches to design efficient privacy-assured searchable encryption schemes, which are based on several recent symmetric-key encryption primitives.

2.8 Conclusions

The motivation for cloud computing has been increasing and the concept connects to the development of the economy. Thus, the growth in number of applications and the amount of data that need to be managed caused the clouds storages to pass to be an important item in spending and public cloud computing seems to be a way to control these costs.

So cloud computing promises to change the economy, But before regulated and sensitive data are migrated to the public cloud, is necessary to address issues relating to safety standards and compatibility covering strong authentication, key management for encrypted data, safeguards against data loss and regulatory reporting [61].

Chapter 3

Data Compression

Given the exponential growth of data that need to be transmitted or stored, and the increasing use of systems connected in networks, the development of applications using multimedia data (text, sound, image and video) that take up much space and difficult the handling, make data compression of extreme importance and increasingly needed, requiring increasingly invest in the best storage and transmission technologies.

The compression is a process that through coding a message reduces the number of symbols required to represent the information contained in the file, to save space and improve performance of data transmission.

The technique of data compression or compression algorithm requires the use of two algorithms, a compression algorithm that receives a set of data and generates a representation of the same data that occupies less space, and the reconstruction algorithm that operates on the generated data by the compression algorithm, and rebuilds the data that was provided initially. Nuno Ribeiro and José Torres [62] refer that data compression converts a stream of input data (original data stream) into another data stream containing the compressed data that occupies less space. The data stream that may be a file or buffer memory, when decompressing becomes an output data stream, which may or may not be identical to the data contained in the original stream, depending on whether a technique of lossless or lossy compression, being compression performed by an encoder and decompression by a decoder. The encoder assigns the code for each symbol or set of symbols, according to information obtained from the model, in decoding is performed the reverse way, this is, the decoder gets codes from the channel and converts it to the corresponding symbols according to the information provided by the model.

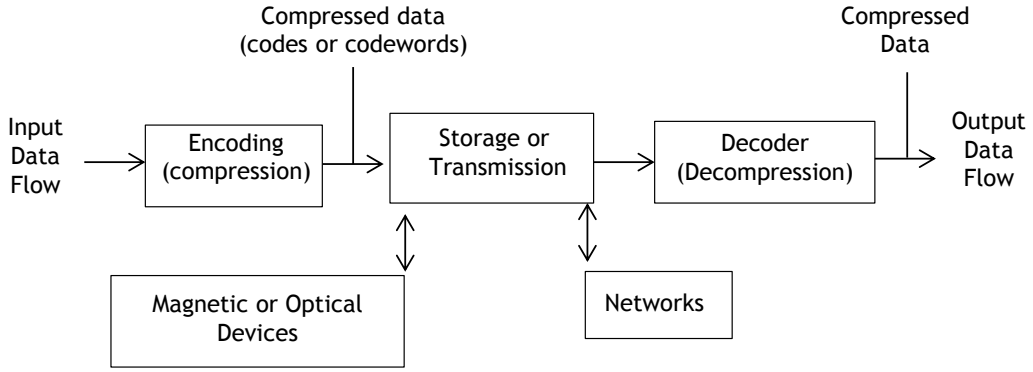

Figure 3.1: Diagram of a generic compression scheme. Adapted from [62].

Nuno Ribeiro and José Torres [62] pronounced regarding to compression ratio referring Salomon [63] that indicates that the performance of a compression scheme can be measured in terms of compression ratio proportioned, and Bhaskaram Konstantinides [64], state that the compression ratio can be obtained in several ways, the most common being obtained by dividing the storage space occupied by the original data stream, or for the space consumed, or for the compressed data stream. Therefore, the compression ratio is equal to divide the original data

stream (before compression) and the compressed data stream (after compression). The higher the compression ratio is, shorter the length of the compressed data stream will be therefore smaller will be the storage space consumed by the compressed data stream, or the lower the bandwidth required for transmission in a computer network. According to the Shannon Coding Theorem (1951), the adequacy of existing information on the model determines the compression ratio.

Compression methods may be divided into two distinct categories, Lossless compression algorithms (lossless) where the end result is equal to the initial and Lossy compression algorithms (lossy) in which the representation of the final data is slightly different from the initial data.

3.1 Lossless Compression

The lossless compression (bit-preserving or reversible compression) is a process in which the information is retrieved without modification, that is, after the decompression process, the decompressed bit stream is identical to the original bit stream. The lossless compression techniques are needed in certain applications where the accuracy of information is essential, for example, compress textual data, numeric data, in the use of images for medical purposes, in files that cannot be altered by the compression process.

3.2 Lossy Compression

Lossy compression (lossy compression or irreversible compression), the uncompressed information is different from the original compressed data, is suitable for files such as digital audio, bitmap images and digital video, consists of an alteration of the original information. Depending on the applied algorithm, the lossy compression of data can lead to generative loss (generation loss), losing more and more information as the compressions are made over other compressions. There are two basic schemes for lossy compression of data:

- In processing codecs, image or sound samples are cut into small segments, transformed into a new space-based, and quantified. The resulting quantized values are encoded.
- In predictive codecs, previous or subsequent encoded data is used to predict the current sound sample or image frame. The difference between the expected and the actual data, with any extra information needed to reproduce the prediction is then quantified and coded.

In some systems the two techniques are combined, with transform codecs being used to compress the different signals generated by the prediction stage.

In most of compression methods, the compression process is more complex than the process of decompression. Therefore, the compression typically requires more processing resources. The algorithms in which this occurs are called asymmetric.

The asymmetric compression adapts to the storage of digital video sequences in video applications on demand or production for distribution of multimedia applications in optical supports, or other storage means.

The advantage of lossy compression methods over lossless compression methods is that normally the resulting compressed file is smaller maintaining, however, a minimum quality of

the original according to the intended goal. The compression ratio (the compressed file size compared with the original or compressed) of video codecs is usually higher than those obtained in sound and images. The sound can be compressed at a ratio of 10:1 (the compressed file occupies one-tenth of the original), without much noticeable loss of quality, as happens with the audio format of MP3 or WMA (Windows Media Audio), with rates up to 320 Kbps of audio (a CD contains audio data to 1411.2 Kbps). But the video can be compressed at a ratio 300:1. Still images are usually compressed at a ratio of 10:1, as in sound but in this case the quality is greatly affected, therefore normally is chosen, for example, a lower ratio 2:1. When a user receives a compressed file with data loss, can be later decompressed but it will be quite different from the original at the level of bits and yet be nearly identical in the normal observation to the human ear or eye. Many compression methods/algorithms rely on limitations of human anatomy taking into account, for example, that the human eye can see only certain frequencies of light. The psychoacoustic model describes how sound can be greatly compressed without noticing the degradation of the quality of the sound signal. The errors/failures caused by lossy compression of data, which are perceptible to the human eye or ear are known as compression artifacts.

A comparison of compression methods implies the existence of metrics to evaluate the performance of the encoding method. According to Buford [65], the four most commonly used parameters are:

- The degree of compression produced by the method of compression, which is translated by the compression ratio, the difference between the input data and the output data. Buford [65] alert to the format specification data input and output to enable a realistic comparison.

- The quality resulting after compression interferes with the compression method used, in terms of quality obtained. This parameter is not suitable for lossless compression since this introduces no changes to the file, however, in lossy compression, the compression result can interfere with the quality of the original data file. In this case we suggest a review to assess whether the loss is imperceptible and does not introduce significant distortions,

- The speed of compression and decompression are two tasks performed in different time, so it should be evaluated separately. In most cases, the decompression speed takes priority, since it interferes with the user's time.

- The hardware and software are two types of resources required for compression and decompression, although it is currently possible to perform compression/decompression by software without having to resort to specific hardware. However the process tends to be slower, and simpler algorithms do not have a good compression ratio. For this reason is better to resort to the specific hardware to speed up the process and permit a good compression ratio. Tends to use the hard-wired.

3.3 Compression Techniques

The data compression techniques can be grouped into two categories:

- Entropy encoding techniques
- Source encoding techniques

Table 3.1: Categories of compression techniques [62].

Category	Main Feature	Compression Method
Entropy Encoding	Independence of the flow characteristics of the data to compress	Lossless Compression
Source Encoding	Takes into account the semantics of the data stream to be compressed	Compression with and without Losses

Nuno Ribeiro and José Torres [62] reported that the techniques of entropy encoding and source encoding are not mutually exclusive. The formats and coding standards for audio, image and video techniques use combinations of the two categories in order to obtain the degree or compression ratio as high as possible.

Several methods and data compression algorithms have been proposed and studied, but some of them deserved remarkable prominence, since they are used in various formats of image files with compression as well as other uses in data compression. The file formats with compression that will be addressed share the use of some compression methods, we opted to treat these methods before entering the scope of the file formats themselves.

3.3.1 Entropy Encoding Techniques

The concept of entropy coding is a term that refers to a technique of lossless coding or lossless compression, that does not take into account the nature of information and ignores the semantics, treats all data as sequences of bits, without optimizing the compression. Nuno Ribeiro and José Torres [62] indicate that lossless compression encoding or entropy refers to methods of compression to which the original uncompressed data can be exactly recovered from the compressed stream. The need for lossless compression arises due to applications requiring lossless compression methods to process sensitive and very important data. Most standards for lossless image compression, audio and video compression follows the global scheme which comprises a combination of JPEG or MPEG, where a lossy technique follows a lossless technique [62], [64].

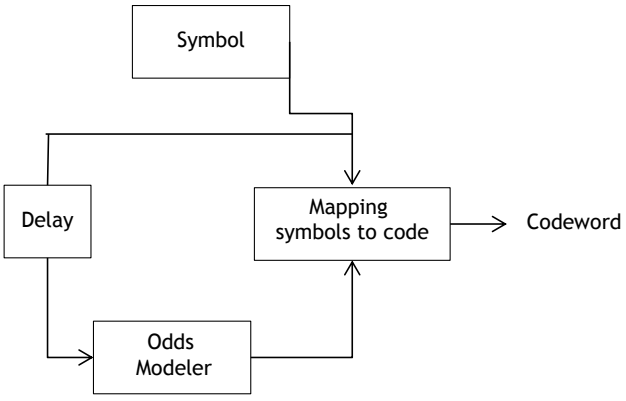

Figure 3.2: Generic Model of entropy coding in accordance with Bhaskaran Konstantinides [62], [64].

Nuno Ribeiro and José Torres [62] concluded that in general, entropy is greater when the probability distribution is uniform and decreases when it has peaks. The entropy principle is to assign short codes for symbols that occur more frequently and the inverse (long codes) for

the symbols that occur less frequently, to obtain an average number of bits in the compressed stream, not smaller than the entropy contained in the original data, thereby preventing loss of information.

The probability model used by the entropy encoder and the decoder can be obtained from the input data stream or from assumptions about the input data. If the model is estimated dynamically from the input data, causality constraints require the presence of a delay function that allows the modeler has all the symbols to determine the frequency of occurrence, if the model is derived from the delay block assumptions is not necessary. The greater the accuracy of the probability model, higher the compression ratio.

The entropy coding techniques can be divided into three main types: suppression of repetitive sequences technique, statistical encoding technique, technique based on dictionaries or redundancy reduction method.

3.3.1.1 Suppression of repetitive sequences techniques

Suppression of repetitive sequences technique is one of simplest and oldest compression techniques in computer science, is based on the output of fixed length codes and operates in two steps, detects repetitive sequences of bytes and replaces these sequences by their number of occurrence.

This technique takes two forms:

- The technique of zero suppression or spaces in which numeric data or zero in on the textual data space is replaced by a special character(a flag or a meta character) followed by the number of occurrences (n) of that character,

- Run-Length Encoding technique is the simplest and is used only to textual content and image data [63] where the sequence of repeated characters is replaced by an abbreviated form. Nuno Ribeiro and José Torres [62] refer in accordance with Fluckiger [66], that "RLE algorithm technique consists in replacing a number of n consecutive characters "c" by the character "c" preceded by a special character (a flag or an escape character) which will be followed by the number n of repeated occurrences of the character. This set of three characters replacing the repeated sequence is called a token, and is represented as follows: !<nxc>". This method should not be used if a character appears repeated only two or three times, that would lead to a sequence longer than the original sequence, it should be used if the number of occurrences of a character is equal to or greater than four.

Thus the Run-length Coding (or RLE) is a technique for compressing strings where there are long sequences of repeated characters. The operating principle of this encoding is simple: When we have the occurrence of a continuous repetition of certain character, for example, AAAAAAAAAAAA, it is possible to replace the representation by the pair (12, A). For the compression of texts this method is not very efficient because depending on the language used will have disadvantages. In the English language, for example, repetitions of the same two letters are even quite common, but repetitions of 3 or more letters are very rare. In the Portuguese language repetitions of 2 consecutive equal letters is rare. This invalidates the use of RLE to compress texts directly. In image compression this technique is more advantageous because the images have higher continuous areas of the same color.

3.3.1.2 Statistical Encoding

Statistical Encoding technique is used for media encoding formats, the type of compression is symmetrical that is the encoder and decoder complexity are identical.

The encoding method is based on statistical identification of byte patterns and encoding each pattern with fewer bits than the number of bits spent to represent the original data stream. According to Salomon [63], in statistical encoding patterns that occur less frequently will be encoded by using more bits, unlike the most frequent patterns that will be replaced with codes less extensive. The statistical encoding method can be implemented by substitution patterns, variable length coding and arithmetic coding.

In statistical encoding the patterns of bytes are replaced in accordance with the frequency with which they occur, the most frequent patterns use shorter codes as opposed of most frequent patterns, therefore the need for the existence of a code table (codebook) establishing the correspondence between the original patterns and their new code either in the encoding or in decoding as in Morse code [66].

- **Coding Method Shannon-Fano Algorithm**

The Shannon-Fano coding is a compression method without data loss prior to Huffman coding that generates variable-length codes for each symbol of the data set to be compressed according to their probability of occurrence. This method was developed by Shannon [67], that developed an investigation in Bells Labs with Robert Fano at MIT [68]. It was described in 1948 by Claude Shannon [69] in his article "A Mathematical Theory of Communication" and attributed to Robert Fano.

The Shannon [67] method is a statistical method that goes through the following steps: file reading and calculation of the frequencies of each symbol; classification of symbols according to their frequencies; frequency dividing into two sub-groups of close frequencies; assigning a code to each symbol; final encoding. The drawbacks of this method are reading the file to calculate the frequency and the need to save code tables.

- **Huffman Coding Method**

The coding technique of David A. Huffman, arises in 1952, is a method of lossless compression with the construction of variable length codes, used in Fax, JPEG and MPEG norms. This method is intended to code with the fewest bits the symbols that the frequency of occurrence is the largest, allows an instantaneous encoding where the first elements of a code word can not be another codeword. This algorithm is based on a statistical analysis of the contents of the file, assign the most frequent symbols, binary codes as short as possible, reserving longer codes to less frequently occurring symbols. This type of encoding requires the creation of frequency tables that have to be transmitted since they are necessary for decoding. This is the most usual technique for the removal of coding redundancy, when applied to the encoding of each symbol of the source individually, Huffman [70] code provides the smallest possible integer number of units of information (bits) per font symbol. The first step is the creation of a series of reductions in the original source, through the ordering of the probabilities of occurrence of the symbols under consideration, combining the (two) symbols with lower probability in a single symbol that will replace them in the next step of source reduction. The second step is to encode, each reduced source, starting with the smallest font and walking toward the original source. The smallest possible binary code to a source of two symbols is obviously formed by the symbols 0 and 1. These values are assigned to the two symbols on the right (in this case, according to the convention 'higher probability receives bit 0'). As the symbol probability of 0.6 was generated from the combination of two other symbols reduced at source to its left, the 0 used to encode it is now attributed to both the symbols that originated it, placing a 0 or 1 to the right of each (according to the same convention) to distinguish them. The process

is repeated for each reduced source till return to the original source. The Huffman algorithm allows the creation of an optimal code for a given set of symbols and respective probabilities, with the caveat that the symbols should be encoded one at the time. The code is called the instant code block and uniquely decodable, because each source symbol is mapped into a fixed sequence (block) bits, each codeword can be decoded instantly, without reference to subsequent symbols and because there is not more than one way to decode a string of 0s and 1s, that is, none codeword is a prefix of any other.

Several investigators note that Huffman coding is good to encode variables with known distributions that have to be coded symbol by symbol. However, due to the fact that the lengths of the codes, have to be integers, there may be a loss of up to one bit per symbol, in the encoder efficiency. We can alleviate this loss selecting blocks of input symbols but the complexity of the approach grows exponentially with the length of the block.

The Huffman algorithm [70] is one of the best known compression techniques. For a given probability distribution is generated a code free of a prefix with minimum redundancy, besides producing a sequence of random bits. Using variable-length codes to represent symbols of the text or characters that can be strings (digramas, trigrams, words or n-grams). The algorithm assigns smaller bit codes to the most frequent symbols in the text, and longer codes to the rarest.

- **Canonical Huffman Coding Method**

The canonical Huffman method is structured to allow rapid decoding requiring only a few bits per symbol of the alphabet.

The canonical Huffman code is generated by an algorithm that takes into account only the size of the Huffman code for the symbol, that is, the height of the symbol in the tree. Visually, the canonical tree is the same as Huffman tree, where all nodes are shifted to the left (or right) within each level of the tree. To build the tree, is just needed to know the size of the Huffman code for each symbol and from there the leaves are positioned at appropriate levels, from left to right, upward sorted by frequency.

Then the tree is completed creating internal nodes. The traditional Huffman algorithm can be used to determine the size of the codes. The process of construction of Canonical Huffman codes is faster and cheaper. The algorithm uses only the Huffman code length to derive the canonical code, and the great advantage is a substantial gain in the decoding process.

Some researchers claim that, Canonical Huffman method follows the following rules:

- The smaller codes have numerically (filled in with trailing zeros) higher values than longer codes,

- Within the same length, the numerical value increases with the alphabet. Thus the longest code contains only zeroes and each code differs 1-bit from the previous one.

- **Adaptive Huffman Coding Method**

The Adaptive Huffman Coding Method avoids the need to calculate the probability in standard Huffman. This method uses balanced binary tree in which the symbols are going to be inserted as they occur in the source code. Since the same tree is generated by the decoder does not need header.

In this method, a special symbol is used which is placed immediately in the tree at the time of initialization, which is the character whose code goes to the file indicating that the character is occurring for the first time. Following this code appears always the new character.

With this information the decoder will be able to generate the same tree of the encoder.

Huffman trees must always obey the following property: Are organized, from left to right and from bottom to top, frequencies are always sorted in ascending order. In that way, symbols with higher frequencies have smaller codes. At each level the frequencies are ordered from left to right to simplify the update of the tree.

The algorithm of Shannon-Fano and Huffman coding use variable length techniques: The most frequent symbols should take codes with fewer bits and vice versa. The best known example is the Morse code, that already used at the time shorter codes for more frequent letters of the alphabet. Reside however some problems that lead to energy losses:

- The change of language, in which the most frequent letters in English are not always the most common in Portuguese,

- Encoding with variable length words wherein we not know if a given sequence of bits form multiple encoded symbols with short codes or symbols encoded with long codes. We must know if the selected code does not contain ambiguities and can be uniquely and instantly decoded.

- **Adaptive Huffman Coding Method**

Nuno Ribeiro and Jose Torres [62] according to Bhaskaran Konstantinides [64], indicate that arithmetic coding is a lossless compression technique that takes advantage of processing many symbols as a single unit of data, while holding the incremental approach encoding symbol-by-symbol (such as Huffman).

Arithmetic coding allows dynamic adaptation of the model probabilities without affecting the design of the encoder. The arithmetic compression algorithm is not based on symbol tables. The arithmetic encoder eliminates the association between individual symbols and code words of full length and equals the entropy of the source from a statistical model, builds up a table which lists the probability of the next symbol read to be one of the possible symbols.

3.3.1.3 Dictionaries based techniques: Redundancy Reduction Method

The compression techniques based on dictionaries replace sequences of symbols, by an indication of its location in a repository called the dictionary of phrases. Practically every dictionary-based method is a variation of LZ77.

The dictionary based coding produces fixed-length codes that are associated with symbol sequences of varying length that occur together. The compression methods based on dictionaries do not need to know the statistics of the data to compress, do not use variable length codes and use frequency symbols of variable length. Many of the programs used in compression are based on this technique.

The technique based on dictionaries or method of redundancy reduction, is based on the selection of sequences of symbols and encoding these sequences like a token, using a dictionary that stores symbol sequences. These methods are asymmetrical meaning that the compression is more complex than decompression. The dictionary can be used allowing the static or dynamic addition and removal of sequences as the message is coded. The most used lossless compression programs usually associate a dictionary-based technique with a statistical technique.

The LZ77 and LZ78 methods are part of this group. LZ77 was one of the data compression algorithms developed by Abraham Lempel and Jacob Ziv in 1977 along with other LZ78 compression algorithm published in 1978.

- **The encoding algorithm LZ78**

The LZ78 algorithm uses a dictionary (called D) to store the strings found in the file, and uses codes (position of the sequences in the dictionary, or even a number assigned sequentially to strings found). When reading a character from the file, we seek in D to see if it is already there. If it is found, we read the next character, concatenate on what we had read, and seek in the dictionary the new sequence, now of two characters. While the sequences are already present in the dictionary, the algorithm continues reading and concatenating. Finally, when we find a character that concatenated with the sequence is not present in the dictionary, we issue the pair output (Dseq, c) where Dseq is the code of the last sequence found and c is the character who "broke" the sequence. After issuing the pair output, we introduce the new sequence terminated c in the dictionary and start all over again.

For the algorithm to work we need to initially enter in the dictionary (usually under code 0) the empty string (an entry representing a string of zero length). Thus, when reading a character c never read before by the program the output pair will be (0, c).

The decoding takes place in a similar way, we start with the dictionary with only the code for the empty string, and we will read pairs (Dseq, c). For each read pair we emit in the output the sequence present in the dictionary with the code Dseq, concatenated with the character c. Then this new sequence is added to the dictionary. Note that the algorithm for maintaining the dictionary (code allocation to sequences) must be the same in coding and decoding, so we do not have inconsistencies.

- **The encoding algorithm LZW**

The LZW (Lempel-Ziv-Welch) is a data compression algorithm, derived from the LZ78 algorithm and based on the location and record of the patterns of a structure. It was developed in 1984 by Terry Welch and is generally used on images that you cannot lose the original definition. In the images, the algorithm reads the pixel values of an image bitmap and elaborates a code table where they represent the repeated patterns of pixels found. The LZW encoder reduces by compression graphic images to 1/3 or 1/4 of its original size.

- **The encoding algorithm LZMA**

LZMA algorithm (Lempel-Ziv-Markov) is an improved version of the data compression algorithms LZ77. Developed to improve the level of compression, keeping high compression ratio and low spending memory. The algorithm has been in development since 1998 and is used to perform data compression without loss. This algorithm uses a dictionary compression scheme somewhat similar to LZ77 and features a high compression ratio (generally higher than bzip2) and a variable compression dictionary size (up to 4GB), maintaining decompression speed similar to other compression algorithms used.

3.3.1.4 Mixed methods: statistical and dictionary

Some researchers suggest that it is possible to use compression methods in an autonomous way in order to monetize renting the effectiveness. It is possible to do it with statistical and dictionary methods, which initially applies the method of dictionary and in a second moment a statistical compression algorithm allowing a second compression. This mixed technique allows to compress 50-65% and is used by many compression tools like ARJ, LHA, PKZIP, UC, GZIP, WINZIP and ZOO.

3.3.2 Source Coding Techniques

The concept of source coding designate compression techniques in which the transformations that occur during compression depends on the type of the original signal. The techniques included in the source coding takes into consideration the properties of the signal to be compressed, based on the characteristics of the type of media to which the data flow belongs to compress.

- In the case of digital audio, audio signals can be exploited by the compression technique to obtain a shorter data flow

- In the case of speech coding, is possible by removing silences,

- In the case of digital video is performed a search for blocks of pixels common to two successive frames of a video stream.

According to Fluckiger [66], the source coding technique is classified according to three distinct types; transform based coding, differential or predictive coding and vector quantification.

3.3.2.1 Tranform based Encoding

The transform based encoding technique comes from discrete cosine transform or simply DCT. When applying the discrete cosine transform, the most significant coefficients are accumulated at the beginning of the vector (or matrix) of data, leaving the rest with very small values and carrying little information. This type of distribution is already enough for a technique for reducing redundancy (such as LZ77 algorithms, LZ78 or LZW) or an optimized encoding (such as Huffman coding or arithmetic coding) produce better results in the original data. However, for always work with a finite precision in numerical representations used, we end up having loss of data. Therefore, even without applying any shape of quantification, the discrete cosine transform compression is a lossy compression. Some forms of quantification normally used with discrete cosine transform are:

- The elimination of less significant components (is determined a threshold value or even a position in the matrix results of the transform, and eliminates or replace these values by 0)

- The integer division of values for a coefficient of fixed quantification (So its possible to use fewer digits, or bits, to represent values)

- The integer division by a/of quantification coefficient matrix (this technique is used by most standards of data compression, as it is more flexible and allows the array to fit desired image quality).

The transform encoding technique is widely used in digital image processing and data compression implies that the original information suffers a mathematical transformation from the spatial or temporal domain for the initial abstract domain that is most suitable for compression, without causing a loss of quality and being a reversible process.

3.3.2.2 Differential or Predictive coding technique

The Differential or Predictive coding technique is based on encode only the difference between the current value of this sample and a prediction of that value, is made the removal of redundancy between neighbouring pixels, extracting and encoding only new information brought by each pixel. This information is usually defined as the difference between the actual

value of the pixel and the predicted value for that pixel. For this reason, this technique is called predictive coding.

3.3.2.3 Vectorial Quantification coding technique

Vectorial Quantification coding technique is an asymmetrical method. Decompression is fast and consists of directly access a table (codebook) to fix the value found with the respective difference. Can be used as a technique with or without losses, constitutes a generalization of the method of replacement of patterns, it applies to compression of text, data, image and audio. In this process, the data stream to be compressed is divided into blocks (vectors), to apply the vector quantification a table is used, which contains, a set of patterns in which each pattern has the same size of the original vector.

3.4 Digital Image

Some researchers suggest that we can distinguish two categories of images, according to the type of data, that are encoded and then stored to allow image reconstruction. Bitmap images in which the file is composed of color data for each point, and vector images files which contain the mathematical description of the elementary figures that constitute the image.

- The bitmap images (raster) are constituted by a set of points horizontally and vertically aligned as rows and columns in a matrix that is a set of points (pixels) contained on a frame, each location having one or more values that describe its color.

- Vector images are representations of geometric entities. The processor have the task of "translating" these forms of information, interpretable by the graphics card. Since a vector is constituted solely by mathematical entities, is easily possible to apply geometric transformations (zoom, expansion ...), while a bitmap image, made of pixels, might suffer such transformations with only a loss of information, called distortion. Pixelation (aliasing) is the appearance of pixels in an image after a geometric transformation (namely the expansion). Moreover, vector graphics allow to define an image with very little information, which makes the files little voluminous too. On the other hand, a vectorial image allows to represent uniquely simple forms. It is true that the overlapping of several simple elements can give very impressive results, not all images can be worked vectorially: is the case of realistic photographs.

Interest in image compression techniques dates back nearly half a century ago and is increasing, thanks to the popularization of multimedia, the extension to geographically distributed systems and many new inventions that require image compression technology to become viable, such as video conferencing, high-definition TV (HDTV), interactive TV and video on demand (video on Demand - VOD).

Image processing requires a very considerable amount of bits and its storage or its transmission at distance is a challenge. The problem is aggravated when working with moving images (digital video) and in both cases we must use techniques and compression algorithms to reduce the amount of data required to represent an image or many images. The image compression techniques seek to remove redundant information in order to store or transmit the same data, efficiently, and type of compression applied can be with or without data loss.

Depending on the application area, the information that is needed to preserve can be objective or subjective. In the first case the compression method should allow for accurate

recovery of the original image data. It is said in this case the process is reversible and that has a lossless compression. Otherwise it is called irreversible and has a compression with losses (lossy), i.e., does not allow the exact reconstruction of the original data. The lossless data compression is applied to images in which the quality and image fidelity are important. Are examples of this type of compression the formats: PNG and TIFF (although some variants have data loss). The lossy compression is used where the portability and the reduction is more important than image quality, without underestimate it. The JPEG format uses this type of compression in images. The GIF format also has a lossy compression, but other than JPEG, which greatly damages the image quality. Thus, the image compression is based on removing redundant information existing in the image and there are two categories of image compression, the category nondestructive where it is possible to reconstruct the original image before compression was performed, and the destructive where the compression characteristics of the images are lost allowing to obtain higher degrees of compression.

In digital image compression, three basic data redundancies can be identified and exploited: coding redundancy, interpixel redundancy and psicovisual redundancy, compression is achieved when one or more of these redundancies are reduced or eliminated.

- Coding Redundancy takes into account the tone of the pixel. If the tones of pixels of an image does not occur with the same frequency (probability) the tones most frequent will be encoded with fewer bits. The average number of bits needed to encode a picture is given by the sum of the number of bits used to represent each tone multiplied by the frequency of that same tone. The average length of the codewords assigned to different values of gray tone is calculated by summing the product of the number of bits used to represent each gray level by the probability of occurrence of that level.

- Interpixel Redundancy of a picture, has to do with the two-dimensional array of pixels used for the visualization and interpretation which should be transformed into a more efficient, but not viewable format. The transformations able to remove the interpixel redundancy are known as reversible mappings, if the elements of the original image can be reconstructed from the transformed data set.

- Psicovisual Redundancy is associated with visual information quantifiable or actual. Its elimination is only possible because the information is not essential for visual processing but results in a loss of quantitative information (quantization). This operation is irreversible and visual information is lost, quantization results in lossy data compression.

3.4.0.4 Digital Image Compression Methods

There are a big variety of compression methods, but only a few commonly used for image compression and those that are free to use, make part of the scope of this work.

1. **Lossless Image Compression**

   The generic system consists of two distinct structures: an encoder and a decoder. The encoder treats the input image and creates a set of symbols. The decoder reconstructs the received image that will be an exact replica. In lossless compression we obtain an image without error, in lossy compression we obtain a certain level of distortion present in the reconstructed image.

   In specific cases of image compression, this must necessarily be lossless, if it appears that image is important as medical images or documents for archiving, where losses are not allowed.

Lossless compression techniques allow perfect reconstruction of the original data, but the rates are low (2:1 to 4:1).

- **Codewords of variable length**

To obtain a lossless image compression is necessary to perform the redundancy reduction of the coding. We can encode the grayscale values using variable length codes, that assign shorter codewords to more probable symbols. In practice, the font symbols to be encoded can be the values of grayscale image or the output of a mapping operation.

- **Huffman Coding**

The best known technique for removing coding redundancy is the Huffman coding [70]. When applied to the encoding of each symbol of the source, individually, Huffman code provides the smallest possible integer number of units of information (bits) per font symbol. The Huffman coding is the most widely used data compression technique to remove redundancy from coding and to change from fixed-length encoding for variable length encoding. Are assigned variable length codes for each symbol, the smallest code symbol is assigned to the highest probability of occurrence in the image and so on, until the largest code symbol is assigned to the lowest probability of occurrence. To make the decoding possible without inserting tables of symbols identification, sequences of bits already allocated can not be repeated at the beginning of the new sequences. With that, the length of the code tends to grow rapidly. The Huffman coding becomes interesting for applications where there is a large imbalance between the occurrence probabilities of the symbols. For applications with nearly equal probabilities, the file size tends to be larger than the original. In practice, most applications fits in the first group with very different probabilities of occurrence. The first step to implement the Huffman coding is to create a table with the symbols present in the code and their probabilities of occurrence. From this table it will be generated a tree structure built from a sequence of operations where the two smallest probabilities are joined at a node to form two branches. Each node already built is treated as a single symbol, with the probability being the sum of all the probabilities of the symbols combined in this node. Then arbitrarily attaches itself to one of the branches 0 or 1 [71] Considering this combined probability, again the two smaller probabilities are joined to form a new node and a new combined probability.

- **Golomb Code**

A set of prefix-free codes that can be used for data compression replacing the huffman code that shows great results for certain probability distributions of coded symbols. The method was developed by Salomon W. Golomb in 1966 [63]. Golomb codes apply to any nonnegative integer number n, and depend on a parameter b which must be previously computed for the code, to be appropriate to the data.

- **Predictive Lossless Coding**

The Predictive lossless Coding, consists of removing the redundancy between neighboring pixels, extracting and encoding only new information brought by each pixel. This 'new' information is usually defined as the difference between the actual value of the pixel and the predicted value for that pixel. The predictive compression techniques operate based on a statistical model of the image. The model is embedded in something

known as a predictor. The predictor estimates the next pixel value in a certain order, based on the pixels that preceded that. The predictor can also be two-dimensional to take advantage of the inherent correlation of bidimensional images. Thus, fewer bits are used to encode the differences than the number of bits used to represent a pixel. The smaller the number of bits used in this quantization, the higher the degree of loss of image fidelity, but higher the compression rate will be. The predictive compressors may be adaptive, making the quantizer and/or the predictor adaptive. This adaptability can be used to achieve higher compression or higher fidelity.

- **Bit-plane Coding**

Bit-plane coding is an image compression technique that exploits interpixel redundancies. This method is based on the concept of decomposing an image of multiple shades of gray into a series of binary images, then compressing each using one of the numerous methods of compressing binary images.

- **Lempel-Ziv Coding (LZ)**

The LZ is one of the most common algorithms used for image compression and data in general. It was created in 1977 by Abram Lempel and Jakob Ziv and since then, many variants of this algorithm have been used in several file formats. This method of compression aims to reduce redundancy coding from the use of data dictionaries and does not generate losses. One of the best known versions of the LZ is called LZW, created in 1984 by Terry Welch. LZW is used in the GIF file format. The LZ version approached in this item is called LZ77. This version uses a mobile window that is called window LZ77 that effectuates the compression and decompression. The sequence of compressed data can contain literal values or commands for expansion. The commands for expansion are pairs length/distance that report the location, in the window LZ77. In the case of text compression, each step a new character is compared with the characters in the window LZ77. If there is no character equal to this new character, then the window is moved to include a new character, and this character is sent for output. If the character already exists in the window LZ77, a new character is caught of uncompressed data. Then it is made a new comparison, but now to see if there are the two characters in the window. This procedure is repeated until the characters obtained does not exist in the LZ77 window. So the last character read is disregarded and a pair distance/length is written off, stating the distance the string is from the start of the LZ77 window, and how many characters compose this sequence. Then the window is shifted by the number of characters of the string to include them and the process is restarted with the next character. As can be seen, for compression, is required two structures, one to sweep the LZ77 window and another to sweep the input data. These two frameworks are complementary and their data are moved together. In the LZ77 algorithm, the larger the window size greater is the compression ratio obtained, but also is higher the algorithm complexity, especially as regards the operations of search in the windows. Therefore, there is a compromise between these two factors that should be taken into consideration. An alternative to the use of larger windows is to use hash tables, which increases the search speed.

- **Arithmetic Coding**

As explained in [72], the methods for encoding variable length (Variable Length Coding - VLC) most used in the coding of the wavelet coefficients are Huffman coding and arithmetic coding. Huffman coding is faster while the arithmetic coding allows greater compression. The first is preferable when hardware resources are limited and time encoding/decoding is the main goal. Arithmetic coding is a bit slower than Huffman coding, but is much more versatile and efficient.

Arithmetic coding replaces the input symbols with a specific code. One bit stream input symbol is encoded into a single floating-point number greater than or equal to zero and less than 1. There are two key points in arithmetic coding: the probability of a symbol and its range limit in coding. The probabilities of the symbols of origin determines the compression efficiency. They also determine the limit of the range of source symbols for the coding process. These range limits are contained within a range of 0 to 1. The limit of the range to the encoding process determines the compressed output.

- **Transform Coding**

The transform coding involves a linear transformation of the input signal and the quantization of the coefficients, this is, of result of the transformation. The purpose of transformation is to change the organization of the pixels. In transform coding generally is divided a given NxN size image into a number of sub-images of size nxn, n <N, which are coded independently. The transformation is said to be one-dimensional when the sub--image is Ixn and bidimensional when the image has size nxn. It is noteworthy that the transformation itself does not perform compression. The function of the processing operation is to get more independent coefficients and order them (in terms of their variances) so that they can be more efficiently quantized. The choice of the transform is not only determined by the correlation between the coefficients obtained. The transformation operation requires memory and computational time for implementation, these parameters are also important for such a choice.

- **Lossless JPEG with Huffman Encoding**

Some studies describe the Lossless JPEG encoding [73],[74], the compression standard for still images ISO JPEG which is popularly known as a compression technique based on discrete cosine transform. However, the lossless operation mode does not use the DCT, but a coding prediction. The Lossless JPEG compression technique uses a form of discrete pulse code modulation[71].

2. **Lossy Image Compression**

The lossy compression techniques include various methods such as discrete cosine transform (DCT), vector quantization, wavelet encoding, neural networks and fractal coding. These methods realize high compression ratios (typically 50:1 or greater) but do not permit exact reconstruction of the original input data, therefore are not accepted for certain applications such as digital medical images, satellite images, seismic data, high fidelity images, executable files and other.

The lossy coding compromise the accuracy of the reconstructed image in exchange for

a higher compression. If the resulting distortion can be tolerated, the increase in the compression can be quite significant. There are two methods for lossy image compression, coding schemes based in the compression of individual samples, pixels and coding schemes based in the compression of blocks of pixels.

- Predictive coding with losses; The lossy compression predictive techniques best known are the delta modulation (DM) and Differential Pulse Coding Modulation(DPCM).

3.5 Digital Audio

The digital sound or digital audio is the digital representation of a sound wave through binary code. The process involves the capture and recording, converting the analog to digital sound is called ADC, (Analog to digital converter) and in reproduction, the conversion from digital to analog is called DAC, (Digital to analog converter). The process of converting analog audio to digital entails a loss but the technological evolution of the conversion process has reached a high degree of accuracy and is not reflected any perceptible distinction to the human ear between analog audio and digital representation.

Audio compression is used to reduce the bandwidth (measurement of the frequency in Hertz of the signal) and/or physical space of the audio file. It is presented divided in two points:

The Codification: transformation of the data stored in a file without compression to a compressed file (within a structure called bitstream). The software that performs the encoding is called audio encoder. There are many encoders, LAME software is one of them.

The decoding analyzes the bitstream and expand it back. The software that performs this process is called audio decoder.

The lossy audio codecs (MP3 (MPEG3), AC3, WMA) code the sound causing a loss of quality to achieve higher compression ratios. The lossless audio codecs (Flac, Shorten, Wavpack), code the sound, compressing the file without changing the quality, generating encoded files that are 2 to 3 times smaller than the original files. Are generally used for radios.

Among the algorithms for sound compression, we highlight the following:

- AC3 (Adaptive Transform Code 3) know as Dolby Digital, developed by Dolby Laboratories for use in filmmaking. The AC3 was first used in Laser Disc A Clear and Present Danger in 1995. It is one of the most widely used standards to encode the signals employed in a distribution scheme employing sound of 6 speakers (5 to midrange/treble and one for bass, the subwoofer). The system became popular after being included in the specifications of the format DVD-Video,

- WMA (Windows Media Audio) is a format from Microsoft Corporation. It was introduced as a replacement for MP3, with characteristics of higher compression but this fact was committed by some independent tests,

-The FLAC (Free Lossless Audio Codec) is a codec of audio compression without loss of information, does not remove any audio stream information, maintaining sound quality. FLAC is designed for an efficient packing of audio data, unlike general algorithms for lossless compression such as ZIP and gzip. While ZIP may compress a file cd-quality audio in 10% or 20%, with FLAC is possible to achieve compression rates of 30% to 50%. The MP3 in 128kbps can reach up to 90% compression, but it eliminates a good amount of detail, resulting in a file of markedly lower quality than the original.

3.6 Digital Video

Some researchers refer that digital video can be stored in memory, in digital supports, ready to be processed and integrated into various multimedia applications. Also says that with digital video direct access is possible, which makes the non-linear editing being simple and that the successive recordings do not degrade the image quality. The digital encryption also facilitates and improves the tolerance of transmission errors.

The video compression technologies serve to reduce and eliminate redundant data to a digital video file can be sent efficiently over a network and stored on computer disks. With efficient compression techniques, it is possible to achieve a considerable reduction in file size, with little or no negative effect on the visual quality. The video quality, however, may be affected if the file size is further reduced by increasing the compression level of a particular technique.

JPEG, MPEG-4 and H.264 are the video compression algorithms which we are going to mention:

- The JPEG (Joint Photographic Experts Group) or M-JPEG is a digital video sequence that consists of a series of individual JPEG images. In each JPEG image of a video sequence may have the same quality determined by the compression level chosen for the network camera or video encoder. The higher the compression level, the smaller the file size and image quality. The main disadvantage is that the algorithm does not use any video compression technique to reduce the data, because it is a complete set of still pictures. The result is that this pattern has a relatively high transmission rate or a low compression ratio for the quality generated, compared to the standards of video compression like MPEG-4 and H.264.

- The MPEG-4 is a licensed standard, requiring users to pay a license fee for the monitoring station. The MPEG-4 operates with low bandwidth applications and applications that require high quality images, capture speed unlimited and bandwidth virtually unlimited.

- H.264, also known as MPEG-4 Part 10/AVC (Advanced Video Coding), is the latest MPEG standard for video encoding. It is expected that H.264 will become the preferred video standard in the coming years. This is because an H.264 encoder can, without compromising image quality, reduce the size of a digital video file by more than 80%, compared with the Motion JPEG format and up to 50% more than the MPEG-4 standard. This means that it will take much less network bandwidth and storage space for a video file, a video quality much higher at a certain speed transmission. In the area of video surveillance, it is very likely that H.264 find the quickest traction in applications requiring high speeds and capture a high resolution, such as surveillance of highways, airports and casinos, where the use of 30/25 (NTSC/PAL) frames per second is the norm. It is also expected that the adoption of H.264 accelerate the adoption of megapixel cameras, since the highly efficient compression technology can reduce file sizes and bit rates generated without compromising image quality. Although the H.264 standard provides saving network bandwidth and storage costs, it requires faster network cameras and monitoring stations.

Chapter 4

Architecture, Prototype and Validation of the Framework for Secure and Efficient Storage of Multimedia Content in Public Cloud Environments

4.1 Introduction

Based on the previous background and in accordance with the objectives established and the defined research problem, it was engineered and designed the development of a framework that allows to perform a set of services in two different cloud environments. It is also provided the obtained results at the level of efficiency.

In this chapter it is described the architecture of the proposed framework, the data modeling of the framework, used languages and tools, illustrative images of the prototype and results of tests obtained with prototype.

4.2 Architecture of the Proposed Framework

4.2.1 The OAuth Protocol

In general, a Webservice is a method of communication between two electronic devices that communicate via the World Wide Web It is a software function that can be used through a network address on the Internet or through the cloud, a service that is always available, or according to the i-Web a Webservice is a component, or a logical unit application, accessible through standard Internet protocols. As components, these services have a functionality that can be reused without worrying about how it is implemented. The Web services combine the best aspects of component-based development and the Web, from which XML can represent data using different protocols.

- **OAuth Protocol (Open Authorization)**

The OAuth protocol (Open Authorization) is an open protocol proposed by Blaine Cook and Chris Messina and is specified in its 1.0 version in the RFC 5849 published in April 2010 [75]. OAuth provides a method for clients to access server resources in the name of the owner of a resource. It also provides a process for end users which allows third party access to its resources contained in the server without this having to share their credentials: the application asks for permission to access the user in question and the user gives consent or not, without having to provide his credentials. This permission will remain active even if the user changes his credentials. This permission given to the application may be removed at any time by the user.

- **Operation of OAuth Authentication**

The authentication through OAuth 1.0 is schematically represented in figure 4.1 and consists of five steps [76]:

1. Login and get the consumer key: in this step, the user will have to authenticate to the server by entering his credentials by doing so, will be given to him a key consumer (Consumer Key) and a "secret", these keys act as a pair.

2. Get Request Token. With the keys obtained in the previous step the application will obtain the Request token that will later serve to get the access token.

3. Obtain authorization from the user at this point, it is presented to the user an interface on the server where he/she can accept or reject that application has access to the server

4. Using the Request Token and Consumer Key to get the Access Token. If the user decides to authorize the application to access the server, the application will be provided with an access token which will enable it as the name implies access to all server functionality.

5. Update the Access Token. The access token needs only to be updated if expires; if this does not happen, it can be used as many times as necessary.

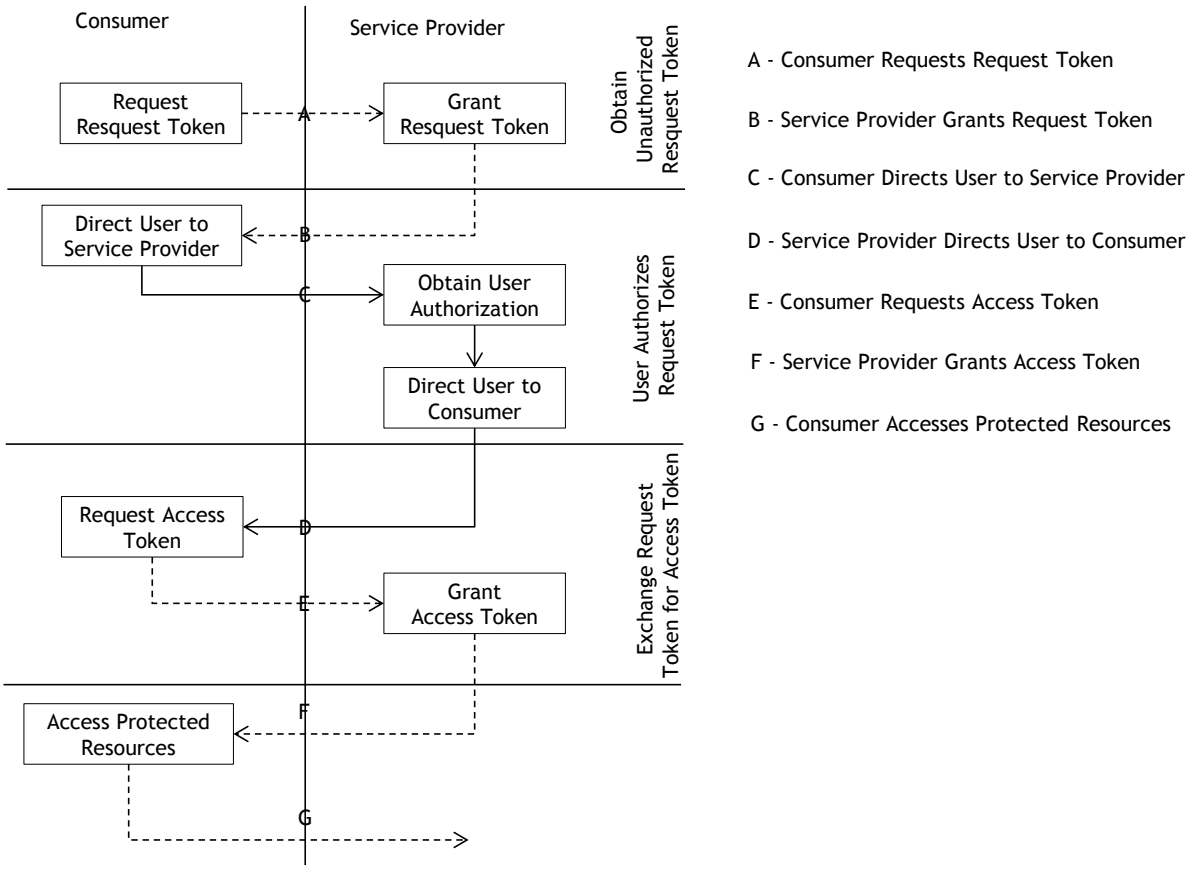

Figure 4.1: Oauth Authentication Flow. Adapted from [76].

One of the dangers of this protocol is a cultural issue, because as the user do not need to provide his/her credentials, he/she feels safe often not realizing that is supplying consent to an application to have access to every single data. If this application is not reliable, it can cause malicious acts. It is therefore important to give authorization only to trusted applications. This protocol is used to generate an access token that allows access to application to data or user files thereby avoiding the need to use the user's credentials.

- **REST (Representational State Transfer) Transmission Method**

The term Representational State Transfer was introduced and defined in 2000 by Roy Fielding [77]. REST (Representational State Transfer) is a style of software engineering for distributed systems such as the World Wide Web. The REST architecture was developed by the W3C Architecture Tecnical Group in parallel with HTTP. REST operates between clients and servers, where clients perform requests to the servers, and servers the servers process customer requests and return responses.

The REST architecture is structured according to the following principles [77]:

- Interaction between client and server: each HTTP message contains all information needed to understand the request made by the client so neither the server nor the client need to record the state of communications.

- A set of well-defined operations that apply to all information resources. The most important ones are: POST, GET, PUT and DELETE.

- A universal syntax to identify resources. In the system REST each resource is uniquely directed by its URI(Uniform Resource Identifier).

- The use of hypermedia which is used for both the application information and the state transitions of the application. As a result it is possible to navigate with a REST resource to many others, simply following the links without having to require the use of records or other additional infrastructure.

Systematizing we can say that REST is a software engineering technique for distributed systems. REST is intended as an image of the application design: a network of websites (virtual state), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and displayed for his/her use. Systems that follow the principles of REST are known as RESTful.

What is important to keep in mind is that the principal in a restful web service are the URLs of the system (usually referred to as restful url's) and resources. A resource is a feature, an entity, i.e., it is an object with information that will be represented by XML. In REST, there is no standard defined, so the operations listed above are just a suggestion. When developing a web service, it is not required to implement all operations to follow a particular pattern, it is just need to implement what is necessary in context. At present it is used in widest sense to describe any web interface that uses XML, HTTP (YAML(Yet Another Markup Language) or JSON(JavaScript Object Notation)) without additional abstractions of standards-based protocols for exchanging messages as SOAP(Simple Object Access Protocol).

4.2.2 Data Structure JavaScript Object Notation for Data Exchange

The format JSON (JavaScript Object Notation) is a lightweight format for computional data exchange. It was originally created by Douglas Crockford, and is described in RFC 4627 [78]. It is a data structure that does not require javascript to work. JSON may replace the XML efficiently. The data structure is simpler to work and the execution time of a script reading data in JSON is faster than reading XML content. One of the advantages claimed by JSON over XML is that is easier to implement.

While JSON is often put on "confrontation" with the XML is not uncommon to see JSON and XML be used in the same application.

The main server side programming languages have support for writing data into JSON. JSON is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd

Edition-December - 1999. JSON is in text format and completely language independent. It uses conventions that are familiar to C languages, including C++, C, Java, JavaScript, Perl, Python, and many others. These properties make JSON to an ideal format for data exchange.
JSON is composed by two structures:

- A collection of name/value pairs. In various languages, this is characterized as an object, record, struct, dictionary, hash table, keyed list, or associative arrays.
- An ordered list of values. In most languages, this is characterized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them, in one way or another. It is acceptable for a data exchange format that is independent of programming language to be based on these structures. There is a growing support for JSON through the use of small packets of others. The list includes the languages: C/C++, C, Java, JavaScript, PHP and others.

4.2.3 Considered Public Cloud Storage Services

The platform has been implemented in Meocloud and DropBox environments, since these environments offer reliable, cheaper and easy-to-use services and to the users. The combination of these different environments in one application allows users to manage their files, with advantages in terms of storage capacity, control costs, data recovery among others. The corporate job and file sharing become easier, once that all information is available in the same application, to where converge all the advantages of the different clouds.

- MeoCloud

The MeoCloud is the new cloud storage service of Portugal Telecom, it is a service of file storage based on the concept of cloud computing. The company provides large central of computers that store the files of clients all over the world. Once the items are copied to the company's servers, are accessible from any place or platform with Internet access. The MeoCloud stores and synchronize files, keeping them safe, available and accessible at all times from any platform or system anywhere in the world. The MeoCloud freely provides 16GB to store your user files.

- Dropbox

Like MeoCloud, Dropbox is a file storage service based on the concept of cloud computing.Dropbox is a freemium service (the customer has the option to use it for free, but also can afford to get some extra functions) of remote storage of files. Files can be uploaded to Dropbox's servers from any device that has an Internet connection, allowing access to files from anywhere in the world. In addition to synchronizing the files, Dropbox keeps old versions of files, allowing the customer, if he/she wishes, to retrieve an older version of his/her data.

4.2.4 Compression and Encryption in the Framework

Through the framework, it is available to the user the possibility to perform the following operations:

- Compress File - In which the user selects a file or several and performs compression

- Encrypt File - In which the user selects a file or several ones and performs encryption, for that the user only needs to insert a password and the framework will perform the encryption using the AES algorithm.

- Compression followed by Encryption - After the selection of files for compression is available an option to the user whether to perform or not encryption after compression.

- Encryption followed by Compression - After the selection of files for encryption is available an option to the user whether to perform or not compression after encryption.

For compression it was used the lossless compression algorithm LZMA and for encryption it was used the encryption algorithm AES, with a symmetric key of 128bits difined by the user.

The choice that the user performs on its priority, compression or encryption has effects in compression as will be shown later.

The chosen method for data compression was LZMA the compression algorithm because it is a lossless algorithm and therefore fall within the scope of this dissertation and because it is an algorithm which has quite good results at the level of compression.

4.2.5 Overview of the Framework Architecture

The platform has been implemented in the environments Meocloud and DropBox, and aims to provide the advantages of combining these storage services in one application. It is also available when using the possibility of transfer and secure storage of the files.

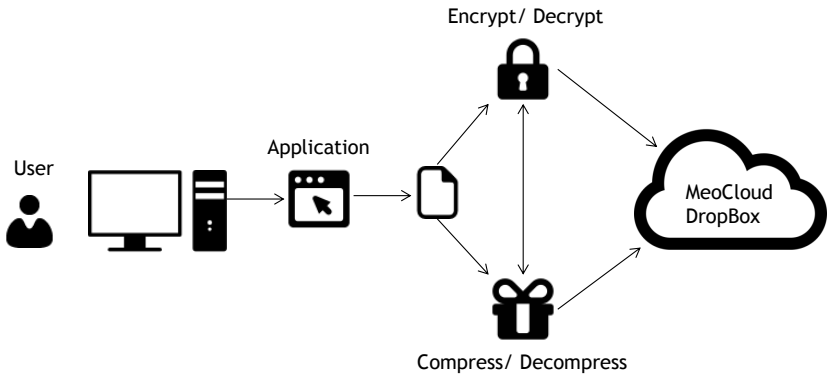

Figure 4.2: Structure and functioning of the Application.

As may be seen in figure 4.2, the user has at his/her disposal in the framework, after performing the login in one or both storage services, the ability to perform file management. The application allows easy access to files with possibility to preview, create folders, delete/download/upload files. It is also available to the user a simple way to take advantage of the maximum storage capacity of cloud storage using the Compress menu. The menu also allows the option to use encryption which in addition to making an even safer transfer of the file, allows to secure the safety of the file when it is stored in the cloud.

4.3 Modeling of the Proposed Framework

4.3.1 Identification of Use Cases

The use case diagrams are a type of classifier representing the functional units, in this case the use cases relate to the application. In table 4.1 are specified functions of the application.

Table 4.1: Use Cases.

Actor	**Uses Cases**
User	Access Control
	View Files
	Download/Upload Files
	Create Folder
	Search Files
	Sort Files
	Move Files
	Preview File
	Delete Folder/File
	Compress/Decompress Files
	Encrypt/Descrypt Files
	Show Shares
	Browsing the Preview
	Logout

4.3.2 Scenarios

The different application scenarios are shown in the different tables (table 4.2 to table 4:29) and refers to the functions performed by the application.

Access Control

Table 4.2: Access Control (Main scenario).

Access Control(Main scenario)	
Precondition	User registration in the system
Description	1 - The use case starts when the application asks the user data (email and password). 2 - The application confirms the user's data and displays a message "Connection established by ...". Allowing too, soon after that, the realization of the logout operation.
Post condition	The user perform the intended operations.

Table 4.3: Access Control (Secondary scenario).

Access Control(Secondary scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the application asks the user data (email and password). 2 - The system does not recognize the data and displays the a message :"Username or Password incorrect". 3 - Then a link is provided to the user that he can use to register in the system: a) The user accesses the link, b) The user registers itself 4 - The user enters data that identifies him in the system
Post condition	The user perform the intended operations

View Files

Table 4.4: View Files (Main scenario).

View Files(Main scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system. 2 - The application provides the files to perform operations
Post condition	The user views and/or perform operations on files

Table 4.5: View Files (Secondary scenario).

View Files(Secondary scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system. 2 - The application contains no files for this user and displays the message "Empty location" providing the option to perform file uploads.
Post condition	The user uploads files.

Download/Upload Files

Table 4.6: Download/Upload Files (Main scenario).

Download/Upload Files(Main scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system and press the button download/upload. 2 - The user is asked to select the file(s) that want to perform download/upload 3 - The user indicates the destination folder. 4 - The user presses the Ok button and a message appears "Download/Upload Started" the user is now able to see a progress bar with the download/upload progress state a) Once the download/upload is completed is displayed the message "Download/Upload completed Sucessfully".
Post condition	The user perform the intended operartions

Table 4.7: Download/Upload Files (Secondary scenario).

Download/Upload Files(Secondary scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system and press the button download/upload. 2 - The user is asked to select the file(s) that want to perform download/upload 3 - The user does not define the files and then press Ok 4 - It is shown the message "Download/Upload Aborted" 5 - It is given the user the possibility to start the operation again
Post condition	The user perform the intended operartions

Table 4.8: Download/Upload Files (Secondary scenario).

Download/Upload Files(Secondary scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system and press the button download/upload. 2 - The user is asked to select the file(s) that want to perform download/upload 3 - The user is asked to set the destination folder for the download/upload 4 - The user does not set the destination folder or enter an invalid path 5 - It is shown the message "Download/Upload Aborted" 6 - It is given the user the possibility to start the operation again
Post condition	The user perform the intended operartions

Table 4.9: Download/Upload Files (Secondary scenario).

Download/Upload Files(Secondary scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user confirms his identity to the system and press the button download/upload. 2 - The application is already occupied performing the download/upload of other files. 3 - A message is displayed asking the user to wait until the application terminates the download/upload
Post condition	The user perform the intended operartions

Create Folder

Table 4.10: Create Folder (Main Scenario).

Create Folder (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user selects the option to create folder. 2 - It is available to the user a window where it is allowed to enter the name of the folder. 3 - The user enters the name of the folder. 4 - The folder is created.
Post condition	The user perform the intended operartions

Table 4.11: Create Folder (Secondary Scenario).

Create Folder (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user selects the option to create folder. 2 - It is available to the user a window where it is allowed to enter the name of the folder. 3 - The user enters the name of the folder. 4 - A folder with the entered name already exists. 5 - The user is asked to enter another name. 6 - If the name is valid the folder is created
Post condition	The user perform the intended operartions

Search Files

Table 4.12: Search Files (Main Scenario).

Search Files (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user places the cursor in the search field. 2 - The user enters a string in the search field. a) The application returns the search result
Post condition	The user perform the intended operartions

Sort Files

Table 4.13: Sort Files (Main Scenario).

Sort Files (Main Scenario)	
Precondition	The user is valid in the system.
Description	1 - The use case starts when the user selects the combobox used for choose for sorting method. 2 - It is shown to the user two options, sorting by size or name. 3 - The user selects one of the options: a) The application returns the files sorted.
Post condition	The user perform the intended operation

Move Files

Table 4.14: Move Files (Main Scenario).

Move Files (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case begins when the user moves a file or folder to another location 2 - The application before moving verifies if exists files with the same name in the destination 3 - The application moves the file/folder. 4 - It is shown the user the message "File Successfully moved".
Post condition	The user perform the intended operations

Table 4.15: Move Files (Secondary Scenario).

Move Files (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case begins when the user moves a file or folder to another location. 2 - The application before moving verifies if exists files with the same name in the destination. 3 - The application notifies the user that exists a file with the same name in the destination and the file in the origin will have the name changed by the application. a) The application changes edit's the file name. 4 - The application moves the file/folder. 5 - The application displays the message "File Successfully moved!"
Post condition	The user perform the intended operations

Preview File

Table 4.16: Preview File (Main Scenario).

Preview File (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case begins when the user presses the button to preview a file: a) The application opens a window and the preview is shown. 2 - It is allowed to the user if the file is a movie, forward, rewind in time increase/decrease the volume
Post condition	The user perform the intended operations

Delete Folder/File

Table 4.17: Delete Folder/File (Main Scenario).

Delete Folder/File (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses the delete button. 2 - It is shown to the user in a message window if he's sure that wants to delete the folder/file 3 - The user confirms the operation. a) - The application deletes the folder/file 4 - It is shown to the user the message "Folder/File deleted successfully!"
Post condition	The user perform the intended operations

Table 4.18: Delete Folder/File (Secondary Scenario).

Delete Folder/File (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses the delete button. 2 - It is shown to the user in a message window if he's sure that wants to delete the folder/file 3 - The user does not confirms the operation. 4 - Message displayed to the user "Operation terminated by user's order" 4 - It is given the user the possibility to resume the operation
Post condition	The user perform the intended operations

Compress/Decompress Files

Table 4.19: Compress/Decompress Files (Main Scenario).

Compress/Decompress Files (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button compress/decompress files. 2 - It is shown to the user a compression/decompression window: - In the window of compression is allowed to add/remove files to a list and browse through folders facilitating the selection of files - In the decompression window is allowed to add/remove files already compressed to list 3 - The user presses the button to compress/decompress 4 - It is shown to the user the message "Compression/Decompression successfully started", is shown to the user also the process state through a progress bar, the application displays when finish's the message "Compression/Decompression performed successfully".
Post condition	The user perform the intended operations

Table 4.20: Compress/Decompress Files (Secondary Scenario).

Compress/Decompress Files (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button compress/decompress files. 2 - It is shown to the user a compression/decompression window: 3 - The user presses the button to compress/decompress a) The application checks if have already a compression/decompression occurring and a message is displayed asking the user to wait for the end of the operation occurring.
Post condition	The user perform the intended operations

Table 4.21: Compress/Decompress Files (Secondary Scenario).

Compress/Decompress Files (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button compress/decompress files. 2 - It is shown to the user a compression/decompression window: 3 - The user cancels the operation, is presented to the user the message "Operation canceled by user".
Post condition	The user perform the intended operations

Encrypt/Descrypt Files

Table 4.22: Encrypt/Decrypt Files (Main Scenario).

Encrypt/Decrypt Files (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button encrypt/decrypt files. 2 - It is shown to the user a encrypt/decrypt window: - In the encrypt window is allowed to add/remove files to a list and browse through folders facilitating the selection of files. - In the decrypt window is allowed to add/remove files already encrypted to a list 3 - The user presses the button to encrypt/decrypt. 4 - It is shown to the user the message "Encryption/Decryption successfully started", is shown to the user also the process state through a progress bar, the application displays when finish's the message "Encryption/Decryption performed sucessfully"
Post condition	The user perform the intended operations

Table 4.23: Encrypt/Decrypt Files (Secondary Scenario).

Encrypt/Decrypt Files (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button encrypt/decrypt files. 2 - It is shown to the user a encrypt/decrypt window: 3 - The user presses the button to encrypt/decrypt. a) The application checks if there's already an encryption/decryption occurring and a message is displayed asking the user to wait for the operation ends
Post condition	The user perform the intended operations

Table 4.24: Encrypt/Decrypt Files (Secondary Scenario).

Encrypt/Decrypt Files (Secondary Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user presses button encrypt/decrypt files. 2 - It is shown to the user a encrypt/decrypt window: 3 - The user presses the button to encrypt/decrypt. 4 - The user cancels the operation, is presented to the user the message "Operation canceled by user".
Post condition	The user perform the intended operations

Show Shares

Table 4.25: Show Shares (Main Scenario).

Show Shares (Main Scenario)	
Precondition	The user is valid in the system
Description	1 - The use case starts when the user access's to view shares 2 - The application shows the shares and allows the user to create, edit or delete shares
Post condition	The user views and/or performs operations on files

Table 4.26: Show Shares (Secondary Scenario).

Show Shares (Secondary Scenario)	
Precondition	The user is valid in the system.
Description	1 - The use case starts when the user access to view shares. 2 - The list is empty but allows users to create file shares
Post condition	The user perform the intended operation

Browsing the Preview

Table 4.27: Browsing the Preview (Main Scenario).

Browsing the Preview (Main Scenario)	
Precondition	The user is valid in the system.
Description	1 - The use case starts when the user access to the preview window file. 2 - The user has at his disposal forward or backward in the previews
Post condition	The user perform the intended operation

Logout

Table 4.28: Logout (Main Scenario).

Logout (Main Scenario)	
Precondition	The user is valid in the system.
Description	1 - The use case starts when the user presses the logout button: a) The application displays a message asking the user if he is sure about the logout. 2 - The user confirms 3 - The application displays the message "Logout successful!".
Post condition	The user perform the intended operation

Table 4.29: Logout (Secondary Scenario).

Logout (Secondary Scenario)	
Precondition	The user is valid in the system.
Description	1 - The use case starts when the user presses the logout button: a) The application displays a message asking the user if he is sure about the logout. 2 - The user does not confirm. 3 - The application goes back to where it was before the user pressed the logout button.
Post condition	The user perform the intended operation

4.3.3 Use Case Diagram of the Application

This section presents the relation that exist beetween elements of the framework that allow its correct operation.

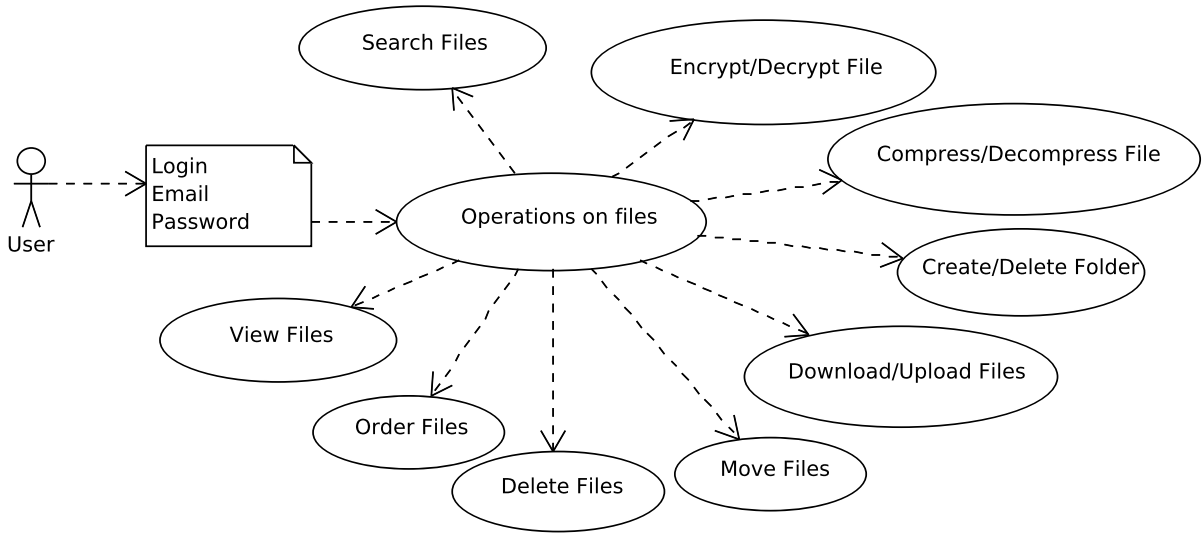

Figure 4.3: Use Case Diagram of the Application.

4.3.4 Class Diagram of the Application

This section presents the various options that the user have to his/her disposal after connecting.

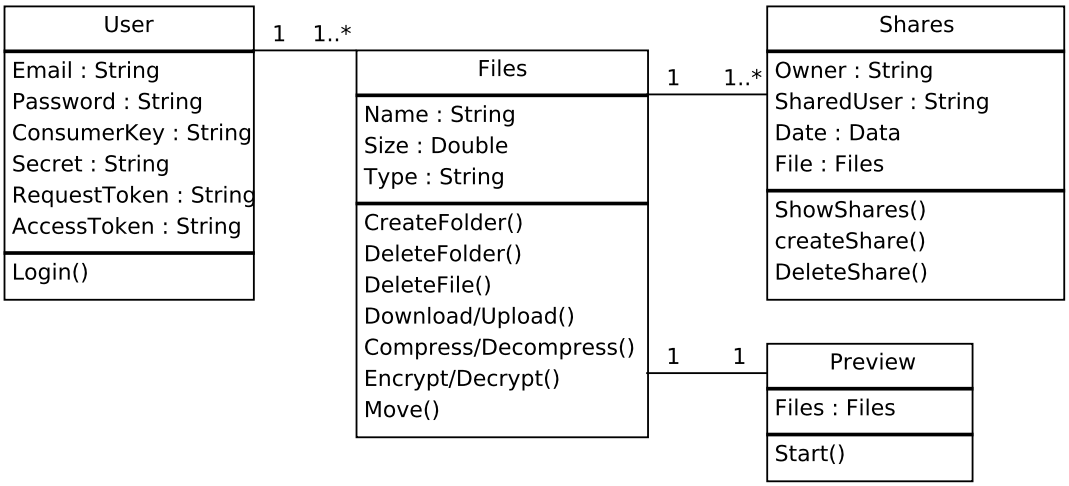

Figure 4.4: Class Diagram of the Application.

4.3.5 Activity Diagram of Application

Activity Diagram - Login

According to the use case "access control", is presented the following diagram activities "Login".

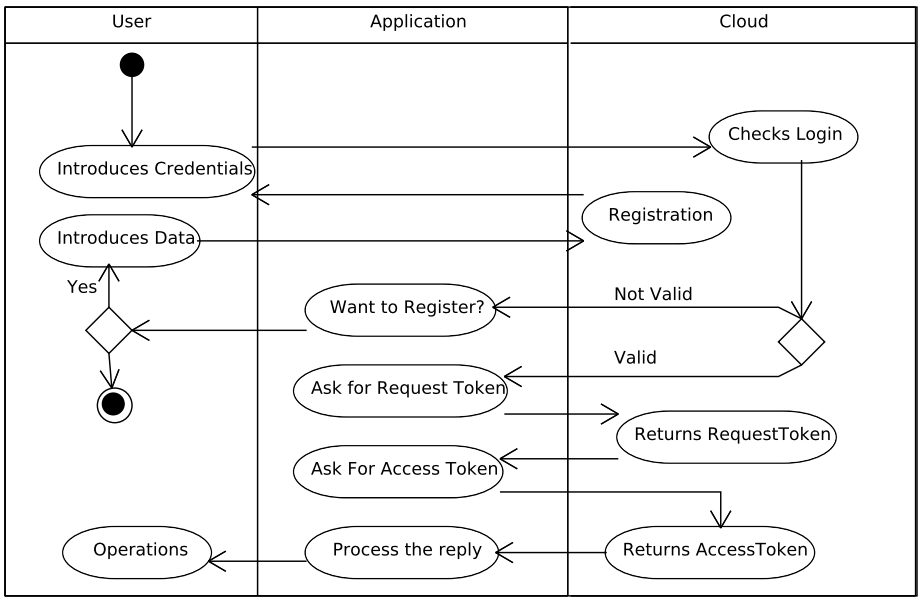

Figure 4.5: Activity Diagram - Login.

Activity Diagram - View Files

According to the use case "View Files", is presented the following diagram activities "View Files".

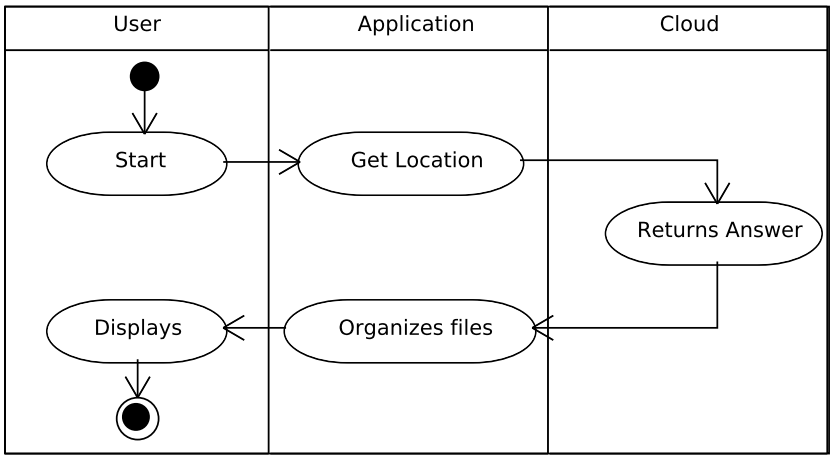

Figure 4.6: Activity Diagram - View Files.

Activity Diagram - Download Files

According to the use case "Download/Upload Files", is presented the following diagram activities "Download Files".

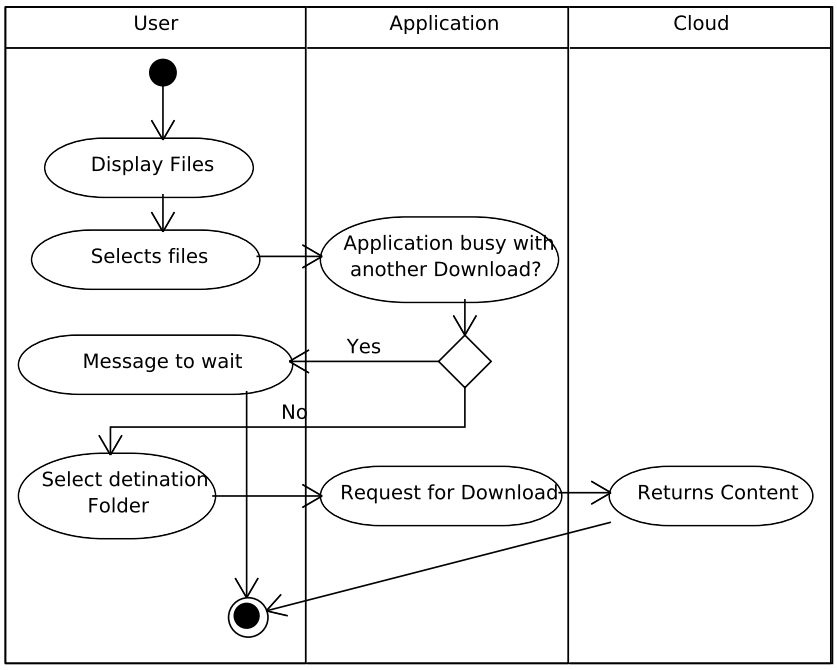

Figure 4.7: Activity Diagram - Download Files.

Activity Diagram - Upload Files

According to the use case "Download/Upload Files", is presented the following diagram activities "Upload Files".

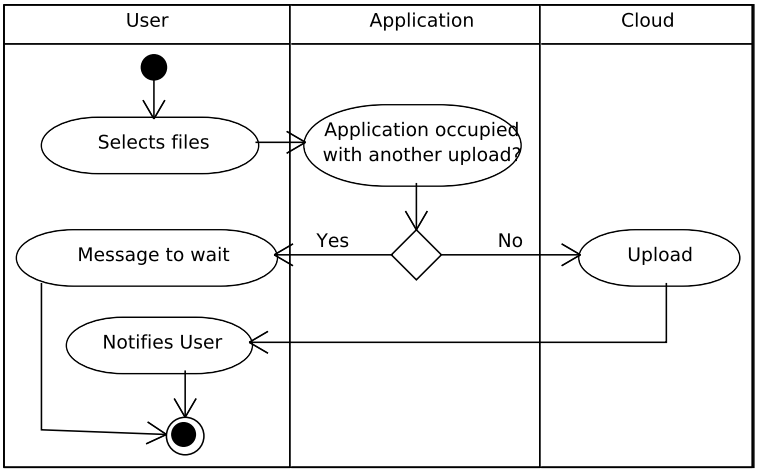

Figure 4.8: Activity Diagram - Upload Files.

Activity Diagram - Create Folder

According to the use case "Create Folder, is presented the following diagram activities "Create Folder".

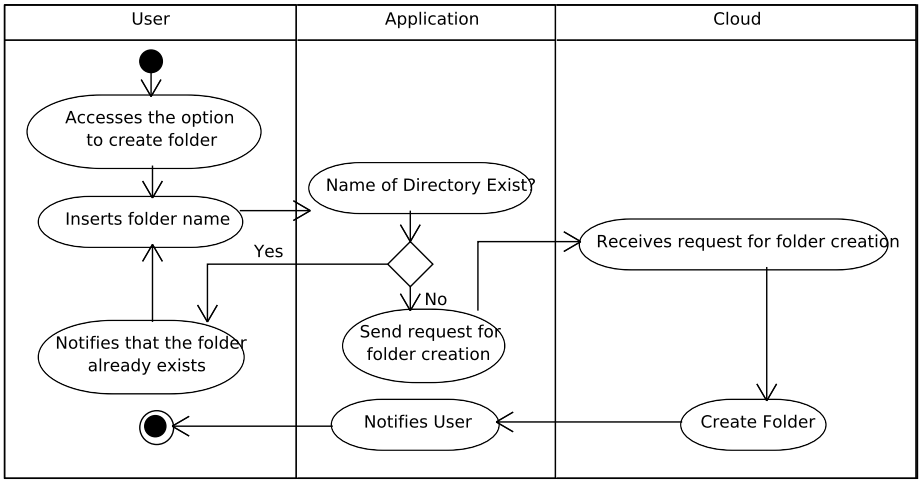

Figure 4.9: Activity Diagram - Create Folder.

Activity Diagram - Delete File/Folder

According to the use case "Delete File/folder", is presented the following diagram activities "Delete File/Folder".

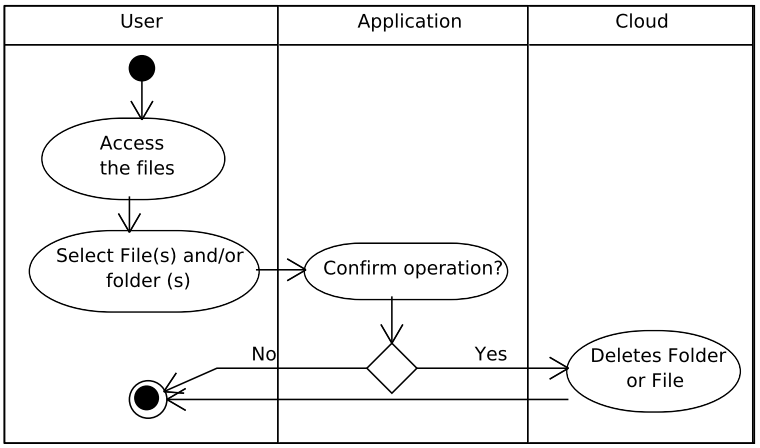

Figure 4.10: Activity Diagram - Delete File/Folder.

Activity Diagram - Search Files

According to the use case "Search Files", is presented the following diagram activities "Search Files".

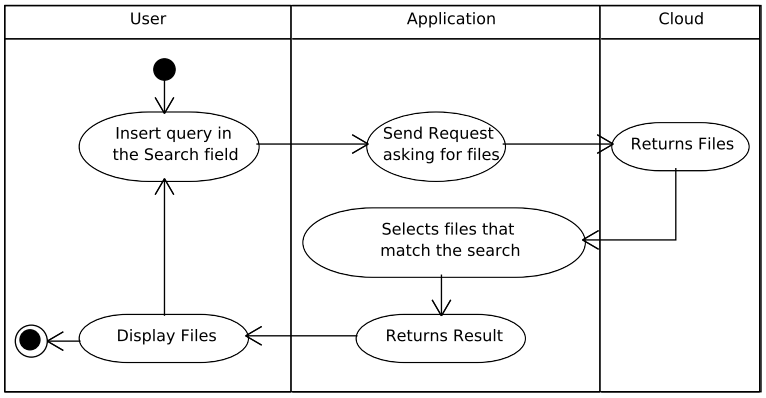

Figure 4.11: Activity Diagram - Search Files.

Activity Diagram - Sort Files

According to the use case "Sort Files", is presented the following diagram activities "Sort Files.

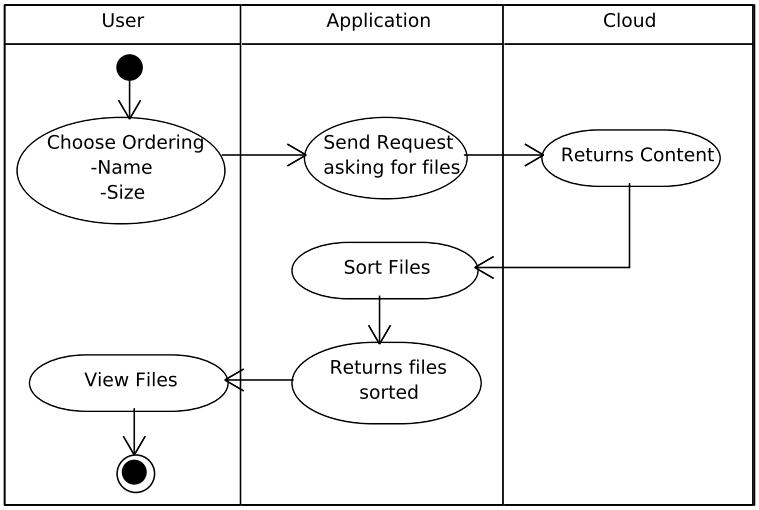

Figure 4.12: Activity Diagram - Sort Files.

Activity Diagram - Move Files

According to the use case "Move Files", is presented the following diagram activities "Move files".

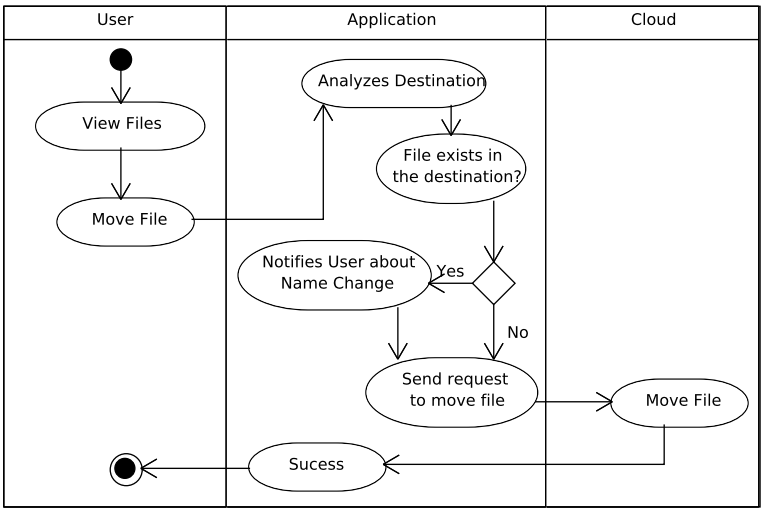

Figure 4.13: Activity Diagram - Move Files.

Activity Diagram - Preview Files

According to the use case "Preview Files", is presented the following diagram activities "Preview Files".

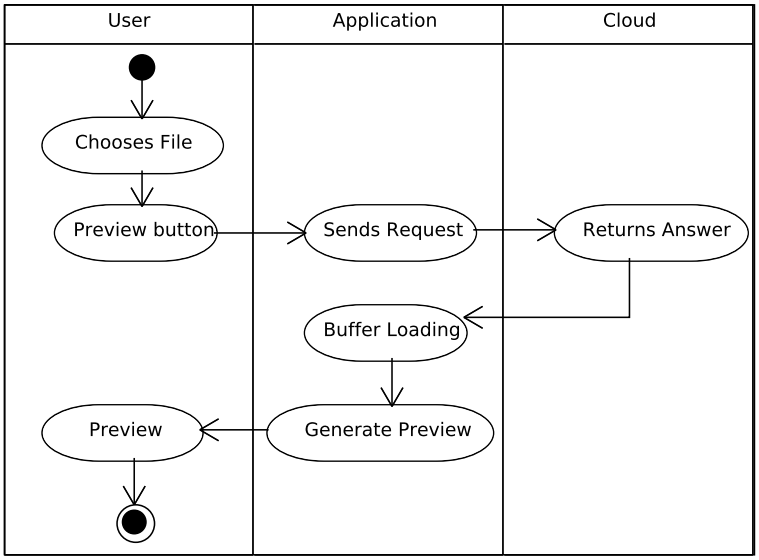

Figure 4.14: Activity Diagram - Preview Files.

Activity Diagram - Show Shares

According to the use case "Show Shares", is presented the following diagram activities "Show Shares".

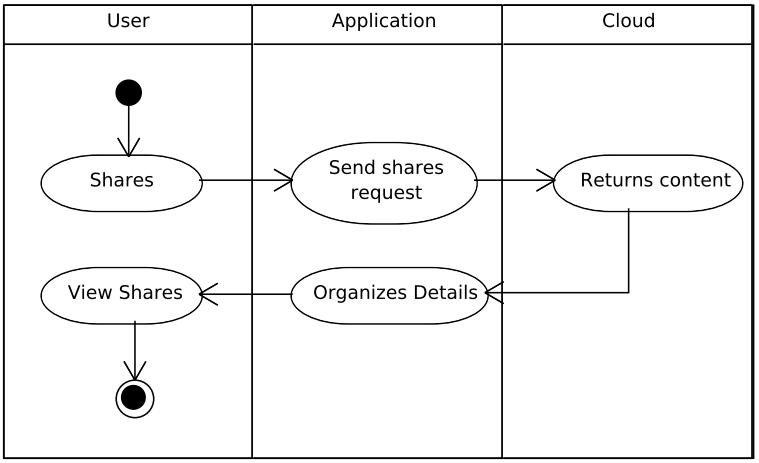

Figure 4.15: Activity Diagram - Show Shares.

Activity Diagram - Create Folder

According to the use case "Create Folder", is presented the following diagram activities "Create Folder".

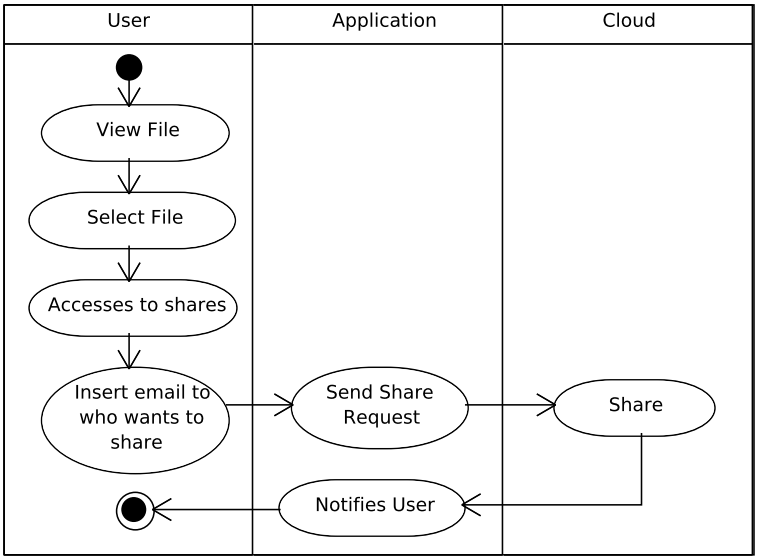

Figure 4.16: Activity Diagram - Create Folder.

Activity Diagram - Compression/Decompression

According to the use case "Compress/Decompress Files", is presented the following diagram activities "Compression/Decompression".

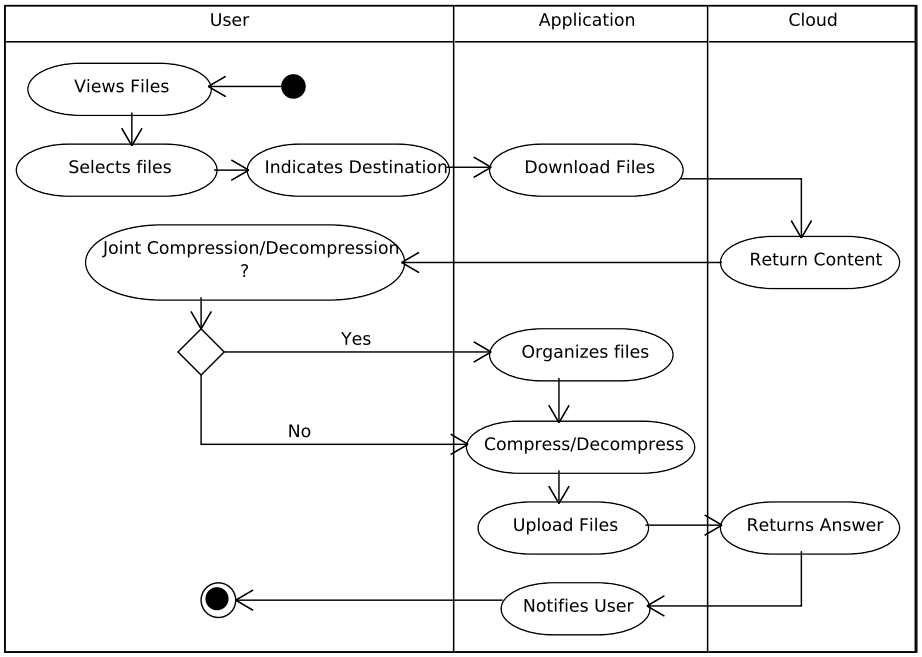

Figure 4.17: Activity Diagram - Compression/Decompression.

Activity Diagram - Encrypt/Decrypt

According to the use case "Encrypt/Decrypt Files", is presented the following diagram activities "Encrypt/Decrypt".

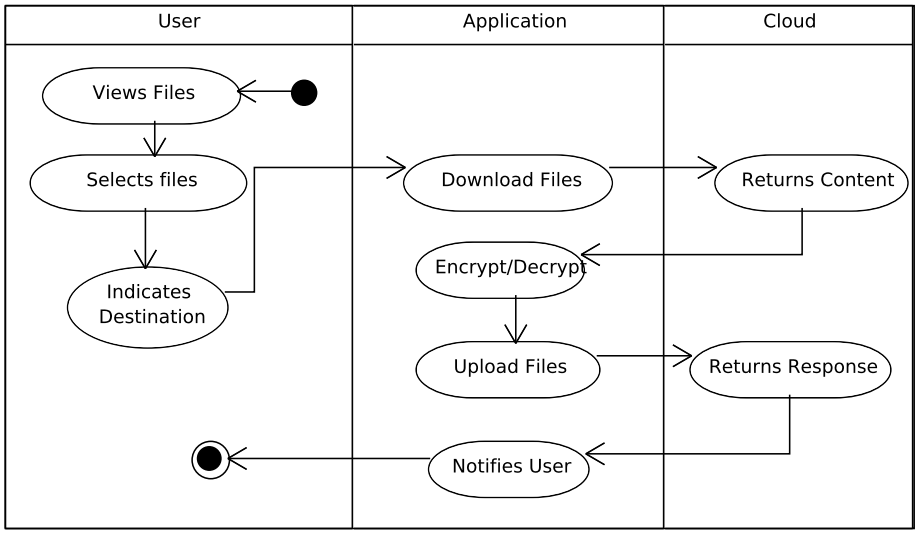

Figure 4.18: Activity Diagram - Encrypt/Decrypt.

Activity Diagram - Compress/Decompress and Encrypt/Decrypt

Based on the use case "Compress/Decompress Files" and "Encrypt/Decrypt Files, is presented the following diagram activities "Compress/Decompress and Encrypt/Decrypt".

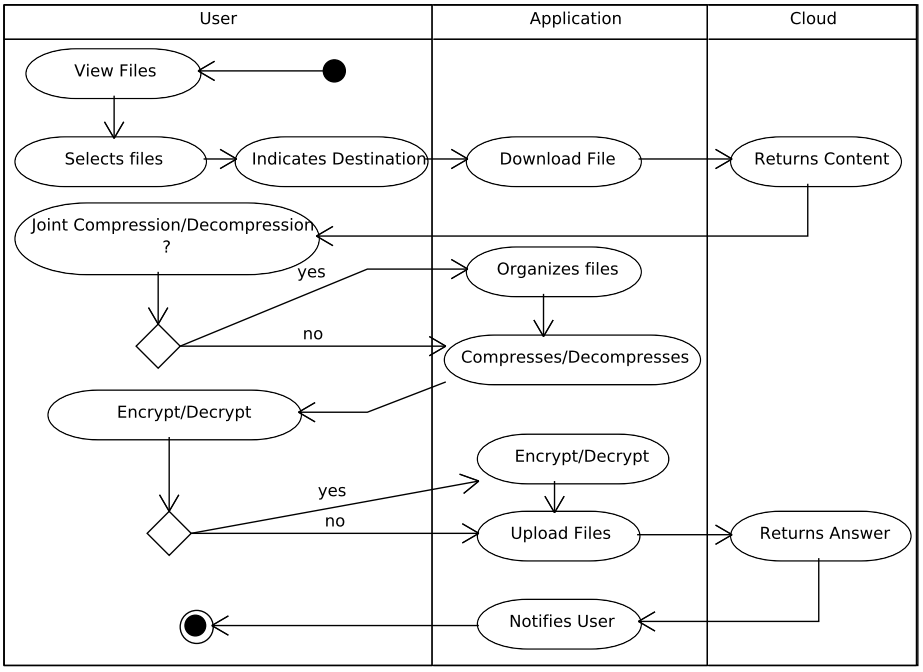

Figure 4.19: Activity Diagram - Compress/Decompress and Encrypt/Decrypt.

4.3.6 Sequence Diagram of the Application

Sequence Diagram - Login

According to the use case "access control", and the Login activities diagram is presented the following sequence diagram "Login".

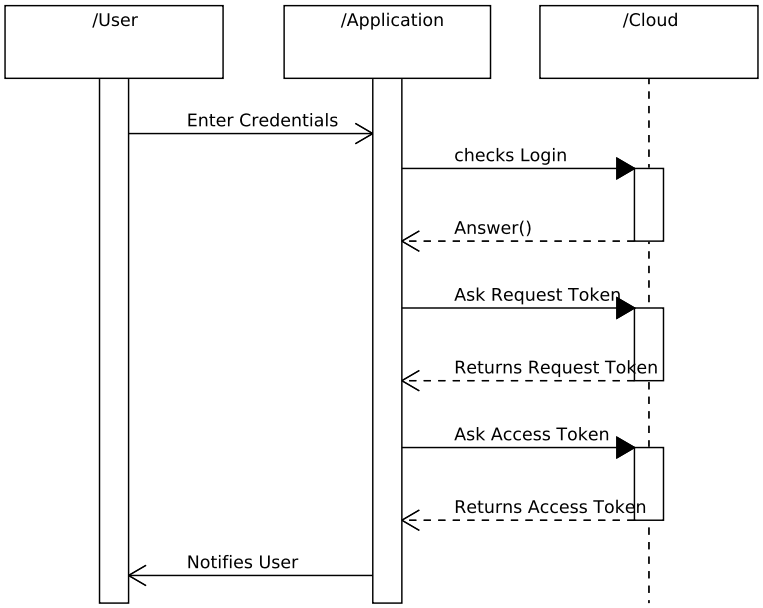

Figure 4.20: Sequence Diagram - Login.

Sequence Diagram - View Files

According to the use case "View Files", and the View Files activities diagram is presented the following sequence diagram "View Files".

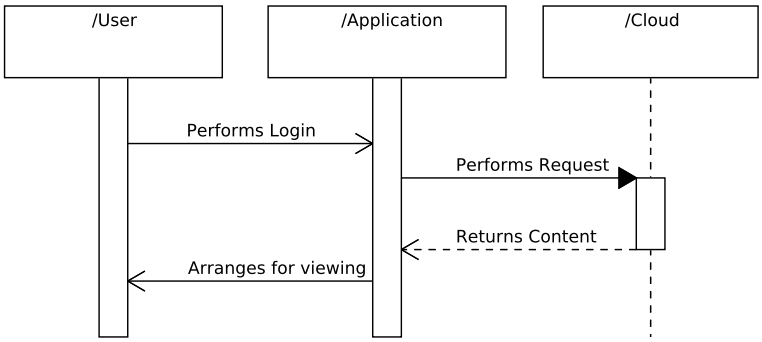

Figure 4.21: Sequence Diagram - View Files.

Sequence Diagram - Download Files

According to the use case "Download/Upload Files", and the Download Files activities diagram is presented the following sequence diagram "Download Files".

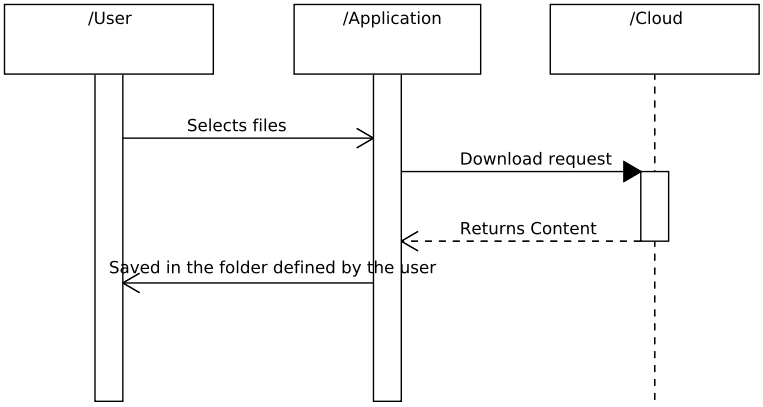

Figure 4.22: Sequence Diagram - Download Files.

Sequence Diagram - Upload Files

According to the use case "Download/Upload Files", and the Upload Files activities diagram is presented the following sequence diagram "Upload Files".

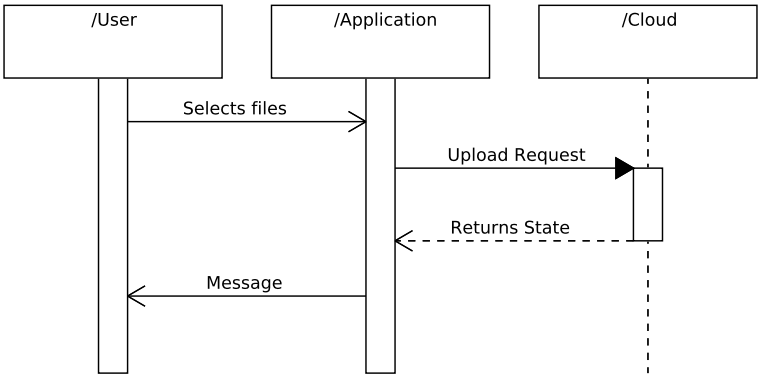

Figure 4.23: Sequence Diagram - Upload Files.

Sequence Diagram - Create Folder

According to the use case "Create Folder", and the Create Folder activities diagram is presented the following sequence diagram "Create Folder".

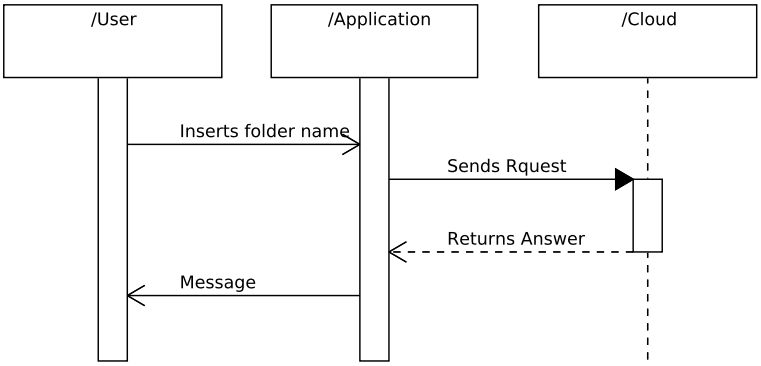

Figure 4.24: Sequence Diagram - Create Folder.

Sequence Diagram - Search Files

According to the use case "Search Files", and the Search Files activities diagram is presented the following sequence diagram "Search Files".

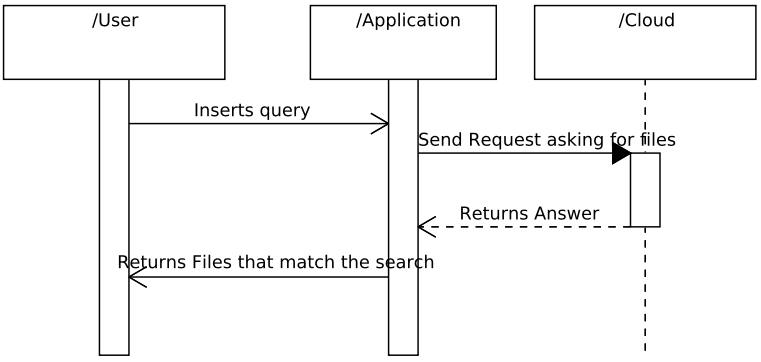

Figure 4.25: Sequence Diagram - Search Files.

Sequence Diagram - Sort Files

According to the use case "Sort Files", and the Sort Files activities diagram is presented the following sequence diagram "Sort Files".

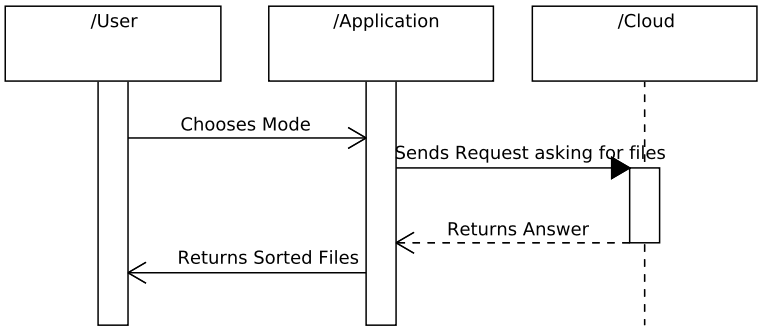

Figure 4.26: Sequence Diagram - Sort Files.

Sequence Diagram - Move Files

According to the use case "Move Files", and the Move Files activities diagram is presented the following sequence diagram "Move Files".

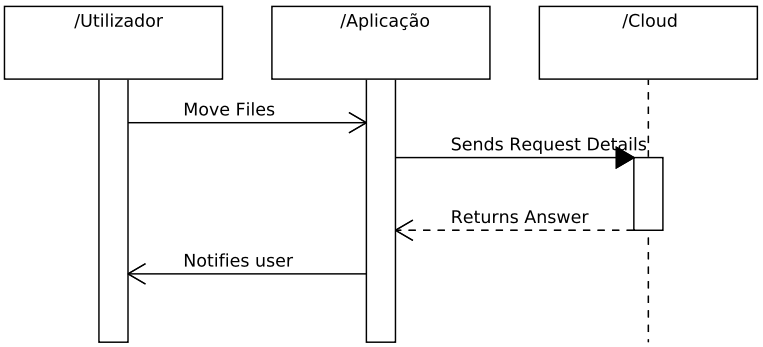

Figure 4.27: Sequence Diagram - Move Files.

Sequence Diagram - Preview Files

According to the use case "Preview Files", and the Preview Files activities diagram is presented the following sequence diagram "Preview Files".

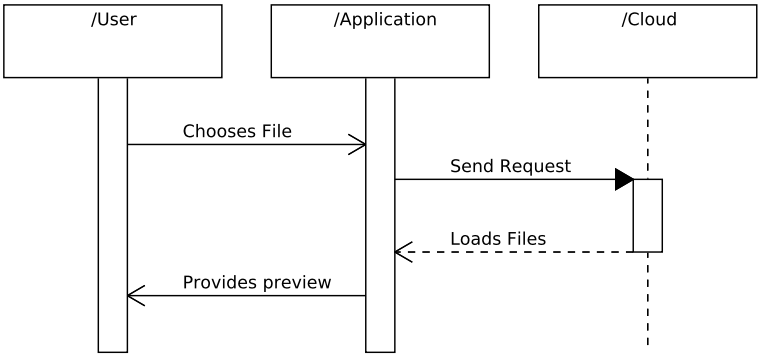

Figure 4.28: Sequence Diagram - Preview Files.

Sequence Diagram - Delete Folder/File

According to the use case "Delete Folder/File", and the Delete Folder/File activities diagram is presented the following sequence diagram "Delete Folder/File".

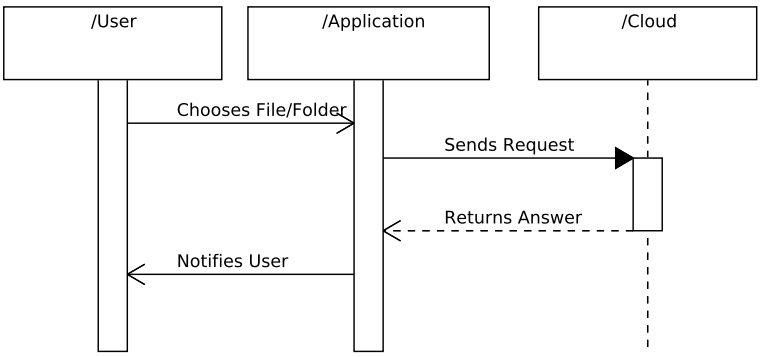

Figure 4.29: Sequence Diagram - Delete Folder/File.

Sequence Diagram - Compress/Decompress

According to the use case "Compress/Decompress Files", and the Compress/Decompress activities diagram is presented the following sequence diagram " Compress/Decompress ".

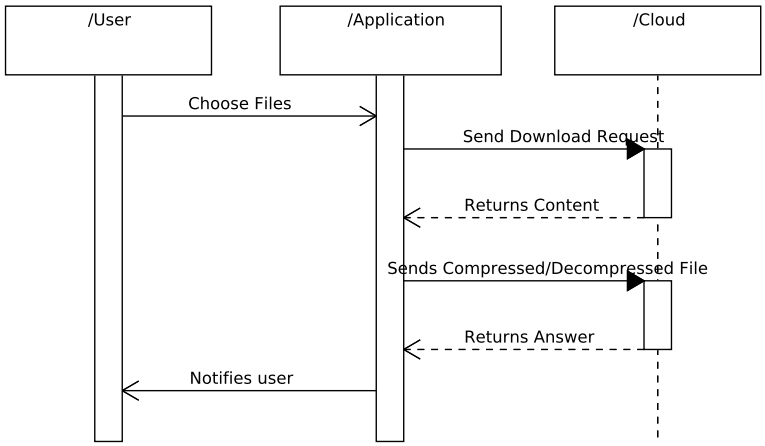

Figure 4.30: Sequence Diagram - Compress/Decompress.

Sequence Diagram - Encrypt/Decrypt

According to the use case "Encrypt/Decrypt Files", and the Encrypt/Decrypt activities diagram is presented the following sequence diagram "Encrypt/Decrypt ".

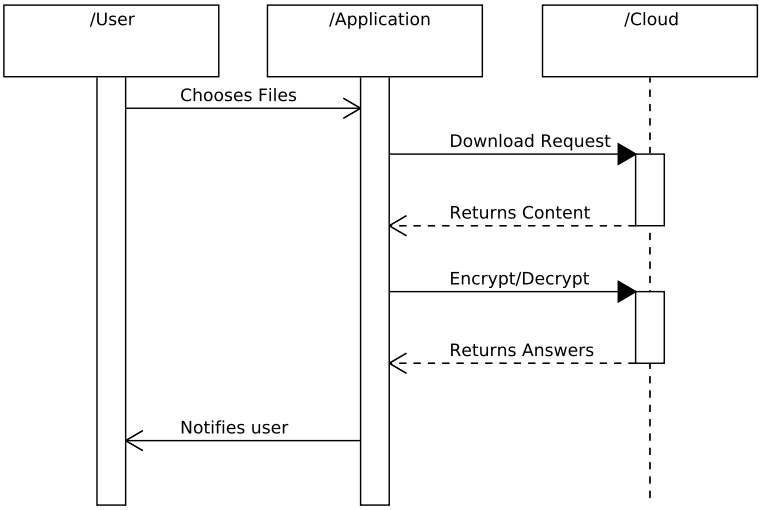

Figure 4.31: Sequence Diagram - Encrypt/Decrypt.

Sequence Diagram - Show Shares

According to the use case "Show Shares", and the Show Shares activities diagram is presented the following sequence diagram "Show Shares ".

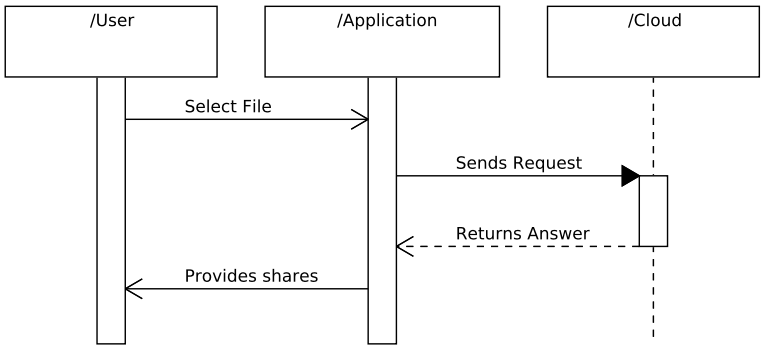

Figure 4.32: Sequence Diagram - Show Shares.

4.4 Languages and Used Tools for the Implementation of the Framework

4.4.1 Used Languages

- **C# (C Sharp) Language**

C# or C Sharp programming language is an object-oriented one which appeared in 2001 and was developed by Microsoft, Anders Hejlsberg and his team, as part of the platform .Net. Its object-oriented syntax is based on C++ but includes influences from other programming languages like Object Pascal and Java. This language is included in the set of tools offered on the platform .Net and emerges as a simple language, robust object-oriented, highly scalable to allow run on various hardware devices, regardless of platform.

It has the following characteristics:

- Allows a new degree of interchange between languages (software components with different languages that can interact). Developers can package up old software to work with new C programs. In addition, C# applications can interact over the Internet using industry standards such as SOAP protocol (Simple Object Access) and XML (extensible markup language)

- It has roots in C, C++ and Java, adapting the best features of each language and adding new capabilities themselves. Provides the features that are most important to programmers, as object-oriented programming, strings, graphics, componnets of user interface (GUI), exception handling, multiple threads of execution, multimedia (audio, images, animation and video), file processing, data structures prepackaged processing, databases, client/server networks based on the Internet and the World Wide Web and distributed computing.

- It is a simple programming language like Visual Basic and powerful as C++, strongly typed to help prevent errors by improper handling of types and incorrect assignments. The memory management programs developed using C# is done by the runtime, via Garbage Collector.

- **JavaScript**

The javascipt was originally developed by Brendan Eich in September 1995 and gained popularity as a scripting language for client-side web pages. Have the following characteristics:

- Supports syntax elements from structured programming language such as C if, while, switch.

- Distinguishes between expressions and statements.

- The line break command terminates automatically, with point-and-optional comma at the end of the command.

- As in most scripting languages, types are associated with values, not variables, for example the variable X could be assigned a number and later linked to a string. The javascript supports several ways to test the type of an object, including ducktyping.

4.4.2 Used Tools

The proposed platform was developed using the Microsoft .Net Framework 4.5 using the programming software Microsoft Visual Studio 2010.

4.4.2.1 Microsoft Visual Studio 2010

Microsoft Visual Studio is a package of programs from Microsoft for software development especially dedicated to the Net Framework and to languages like Visual Basic, C, C++, C# and J#. It is also a great product in the web development using the ASP.NET platform. Was launched with the goal of being the most complete IDE available. For it has support for development of Web applications, applications for Windows Phone, SharePoint while improving the already known, Windows Forms, Web Forms, and also platforms like Microsoft XNA. Offers IntelliTrace, Management Application Lifecycle, a new interface developed with WPF (Windows Presentation Foundation), to make the IDE more intuitive, search system more efficient, among others.

4.4.2.2 Microsoft .Net Framework 4.5

Microsoft. Net Framework as the name indicates is an initiative of the company Microsoft, which seeks a unique platform for the development and implementation of systems and applications. Any code developed for .Net can run on any device that has a framework for such a platform. As in java, the programmer stops writing code for a system or particular device and starts writing for a platform .Net. The .Net runs on the CLR - Common Language Runtime Environment (Independent Execution Language) interacting with a unified set of libraries (framework). This CLR is capable of performing, currently over 33 different programming languages interacting amongst themselves as if they were a single language. The .Net is based on the principles used in java technology (Just-in-Time Compiler JIT) programs developed for it are compiled twice, once in the distribution that generates the code that is known as bytecode and another in the execution. Summarizing the program is written in a language in one of the thirty-three available for the platform, the source code is generated by the programmer and is then compiled by the language chosen generating intermediate code in a language called MSIL (Microsoft Intermediate Language). This generates a new source code file in low-level language Assembly according to the type of project. At the time of execution of the program is recompiled, this time by the JIT compiler, according to the use of the program. The fact that this architecture using MSIL generates a possibility not desired among software developers that is reverse engineering ie it is possible from the compiled code to retrieve the original code. For

this there are tools for code obfuscation to hinder the work of those who attempt to reverse engineer.

This platform enables the execution, construction and development of Web Services (Web applications) in an integrated manner and unified.

4.5 Prototype of the Framework

Authentication

The user can simultaneously access the clouds environments available after registration as is seen on figure 4.33 and 4.34, on the websites of the respective clouds.

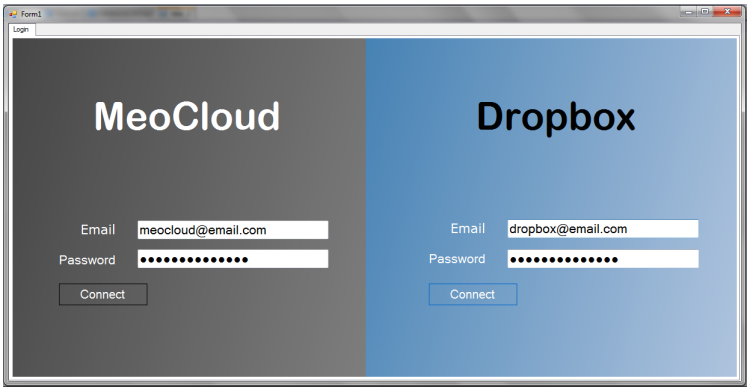

Figure 4.33: Cloud Computing Application: Login Menu.

After entering the credentials (email address and password), application confirms that the connection was successfully established or met any problem (if the user is not registered should proceed to the registration). After the framework confirms that the realization of the connection was successfully established, it is available next to the login tab a new tab (with the name of the cloud service where the connection has been established) where the user can immediately access its content.

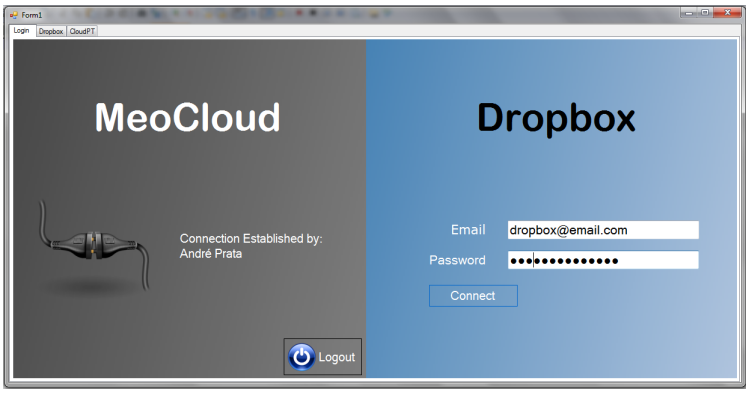

Figure 4.34: Cloud Computing Application: Meocloud and DropBox.

At this point we can see in the picture 4.34 that the link with the Meocloud already been established and the connection with DropBox is ongoing. The link with Meocloud was already been successfully established therefore have become available to the user the option to Logout

whenever he want. As mentioned near the Login tab we now have a tab called Meocloud that allow the user to access all content of this service.

Meocloud

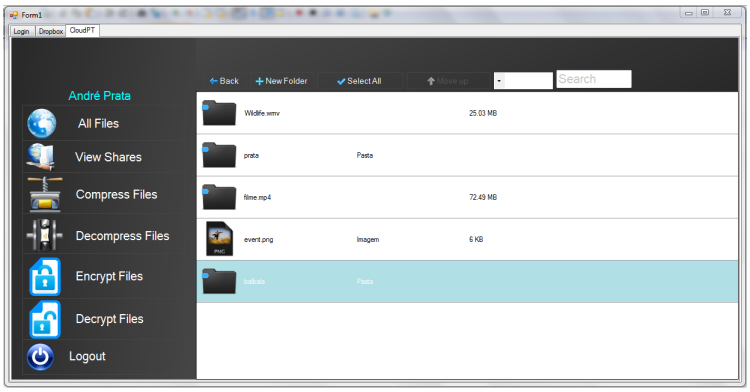

Figure 4.35: Cloud Computing Application: Meocloud.

Mechanisms/Features of Meocloud

The application is implemented resorting the architectures described earlier. The user can as it is referred on the website of Meocloud (www.meocloud.pt), transmit its files securely without losing quality and reliability. Allows the user beyond what has already been described to manage and organize his files.

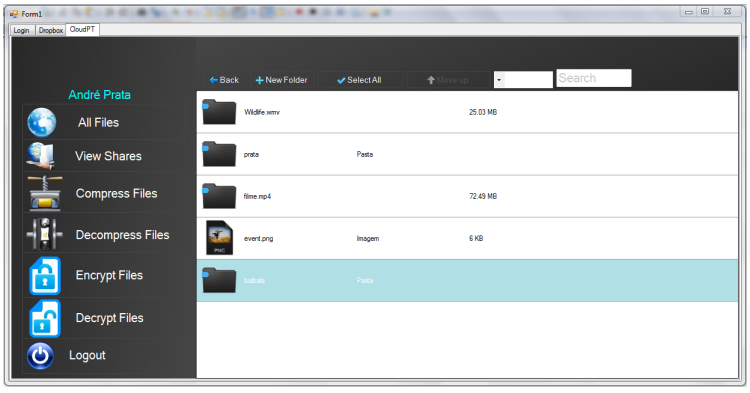

Figure 4.36: Cloud Computing Application: Meocloud.

The application also features means/functionalities that allow the user to:

- Create/delete folders (Figure 4.37) - to create a folder will be displayed to the user a window in which he must enter the name of the folder. After entering the name the application will check if the folder already exists or not.

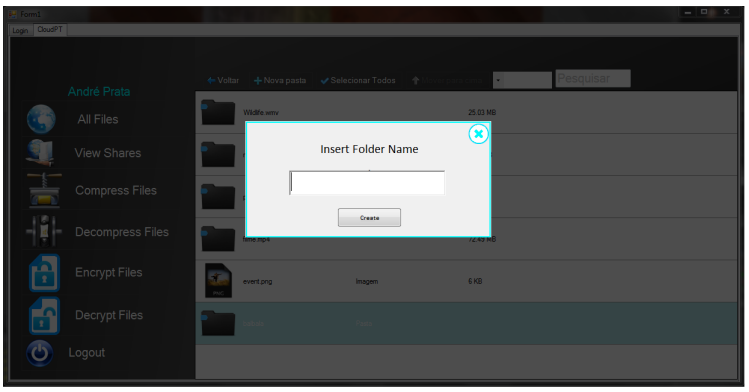

Figure 4.37: Cloud Computing Application: Meocloud Create Folder.

- Upload/download files,

- Search files (Figure 4.38) - user can perform a search by name on the files the user enters a query into the search field, the application will delete files that do not match the text entered.

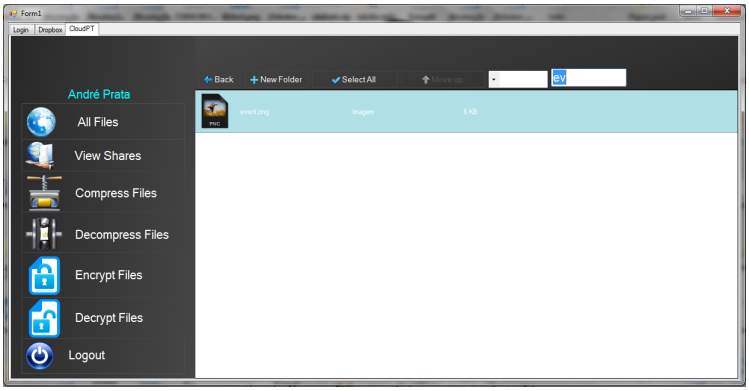

Figure 4.38: Cloud Computing Application: Meocloud Search File.

- Move Files (Figure 4.39) - the user when select a file and dragging it its given the option to change the file location.

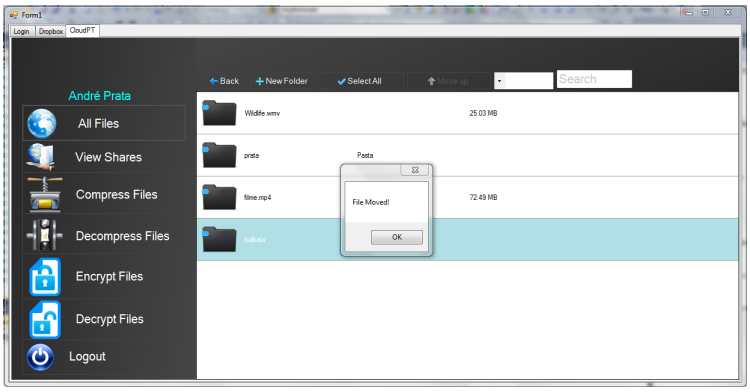

Figure 4.39: Cloud Computing Application: Meocloud Move File.

- Navigate between Multimedia Content (Figure 4.40) - when using the preview files the user is allowed to navigate through previews of files in the folder in question.

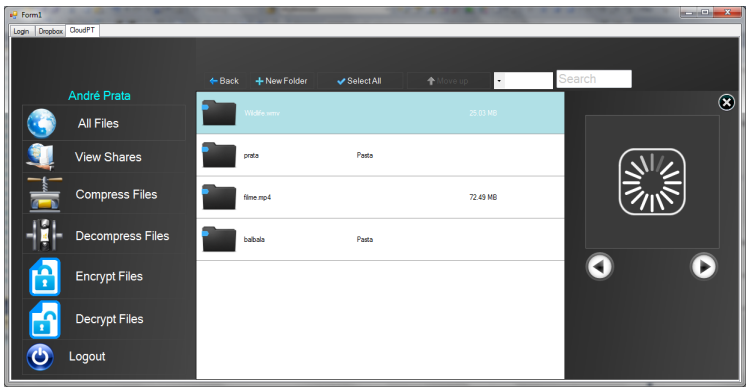

Figure 4.40: Cloud Computing Application: Meocloud Navigate between Multimedia Content.

- Preview the file (photo, video, or other) (Figure 4.41). The user can perform a preview of a file if this is a media content. It is still allowed if the file is a movie, fast forward, rewind time, increase and decrease the volume.

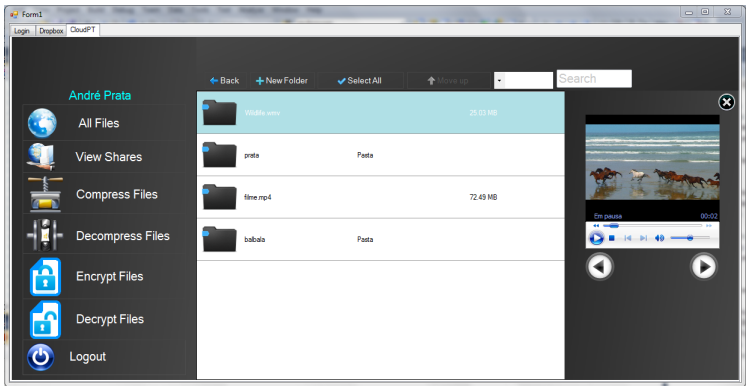

Figure 4.41: Cloud Computing Application: Meocloud Preview the file.

- Sort files by name or size (Figure 4.42).

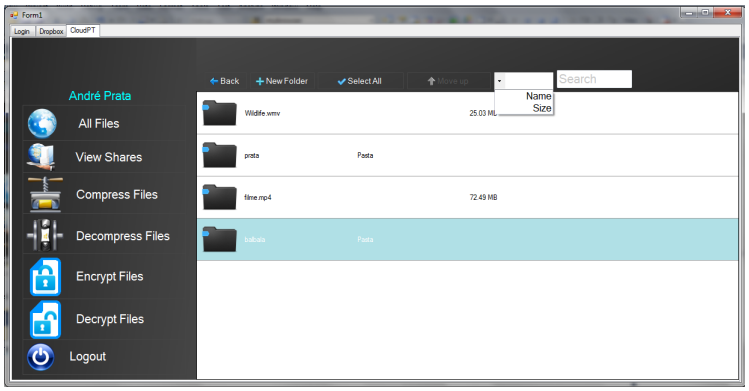

Figure 4.42: Cloud Computing Application: Meocloud Preview the file.

- Compress File Figure(4.43).

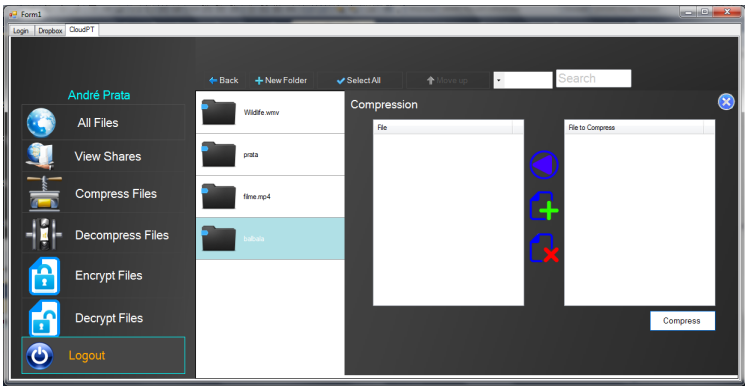

Figure 4.43: Cloud Computing Application: Meocloud Compress File.

- Decompress File Figure(4.44).

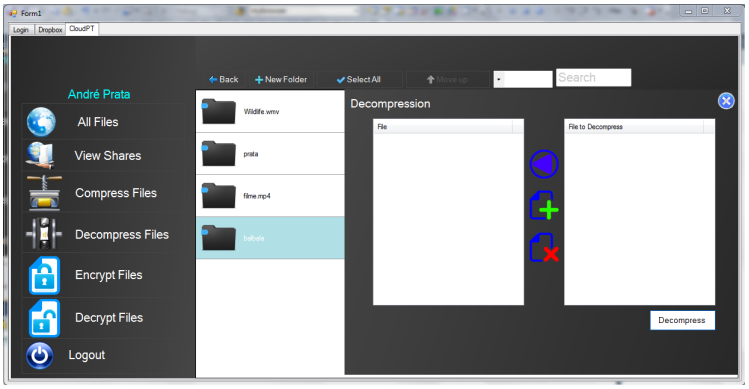

Figure 4.44: Cloud Computing Application: Meocloud Decompress File.

- Encrypt File Figure(4.45).

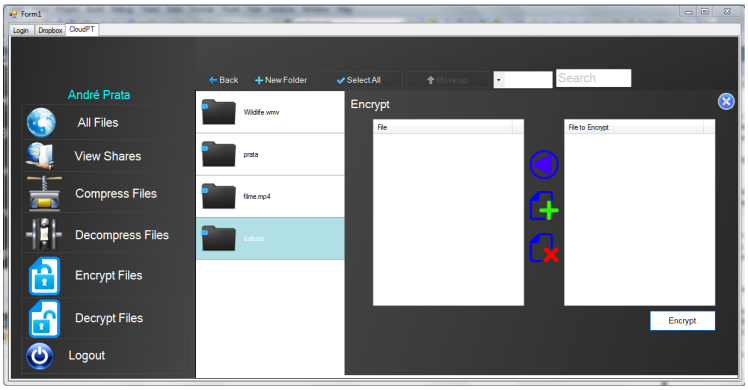

Figure 4.45: Cloud Computing Application: Meocloud Encrypt File.

- Decrypt File Figure(4.46).

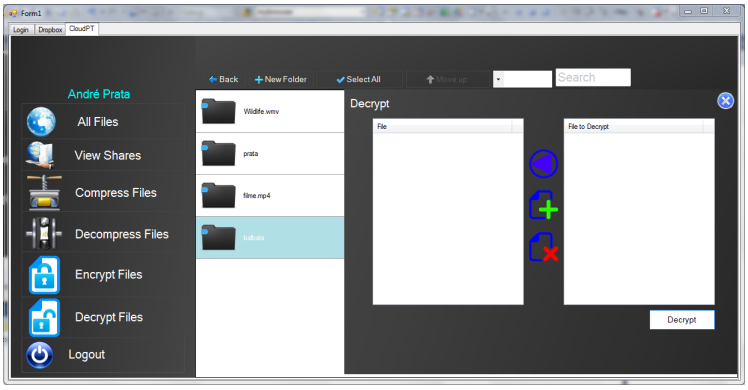

Figure 4.46: Cloud Computing Application: Meocloud Decrypt File.

Dropbox

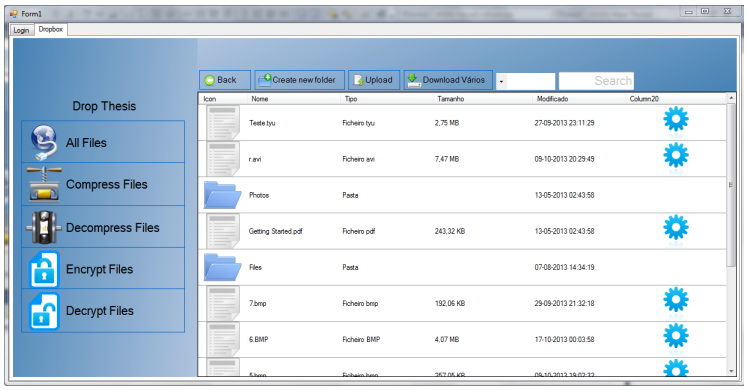

Figure 4.47: Cloud Computing Application: Dropbox.

Mechanisms/Features the DropBox

Was implemented in the application the public API DropBox that provides to the application the ability to perform all operations that performs DropBox on his website (www.dropbox.com). These operations include:

- Create/delete folders (Figure 4.48).

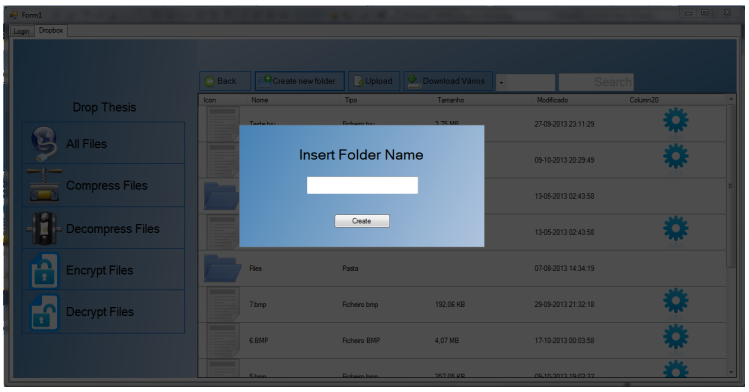

Figure 4.48: Cloud Computing Application: Dropbox Create Folder.

- Uploading/Downloading files (Figure 4.49 and 4.50).

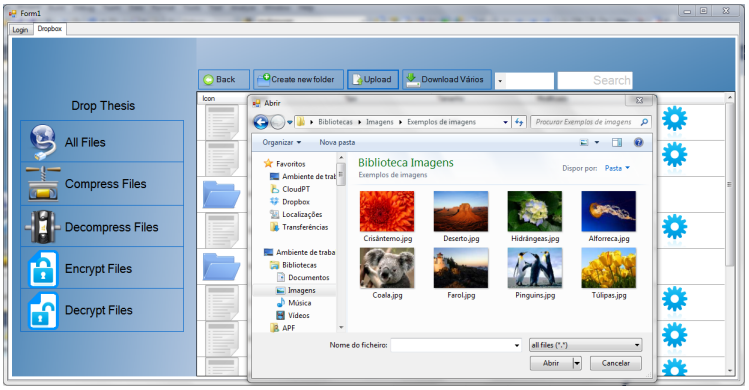

Figure 4.49: Cloud Computing Application: Dropbox Upload File.

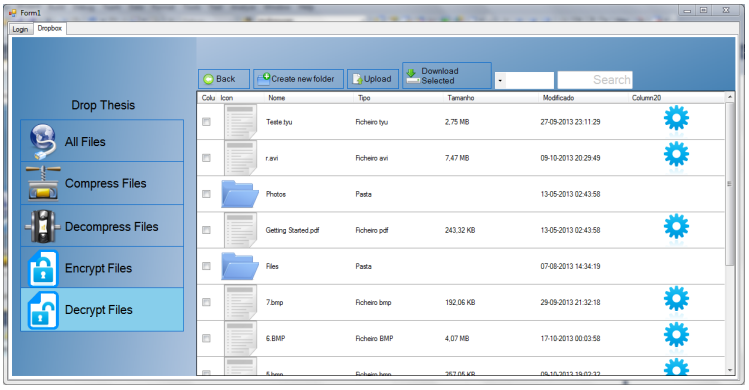

Figure 4.50: Cloud Computing Application: Dropbox Download File.

- Search files (Figure 4.51) - user can perform a search by name on the files the user enters a query into the search field, the application will delete files that do not match the text entered.

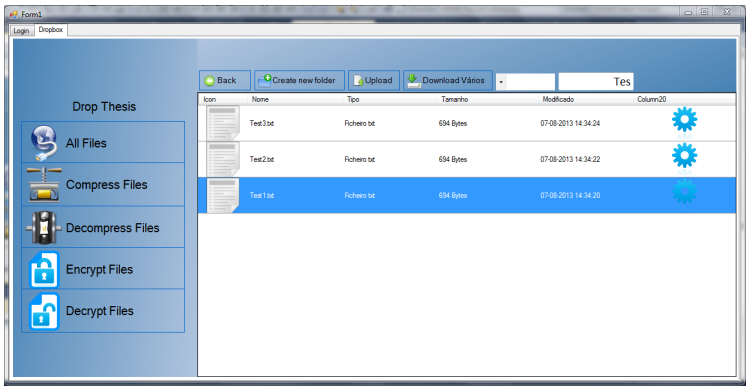

Figure 4.51: Cloud Computing Application: Dropbox Search File.

- Moving files (Figure 4.52) - the user can select a file and drag it to change its location.

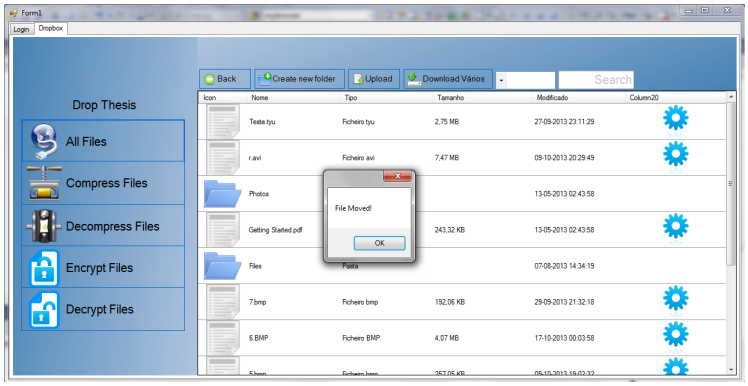

Figure 4.52: Cloud Computing Application: Dropbox Move File.

- Sort files by name or size (Figure 4.53).

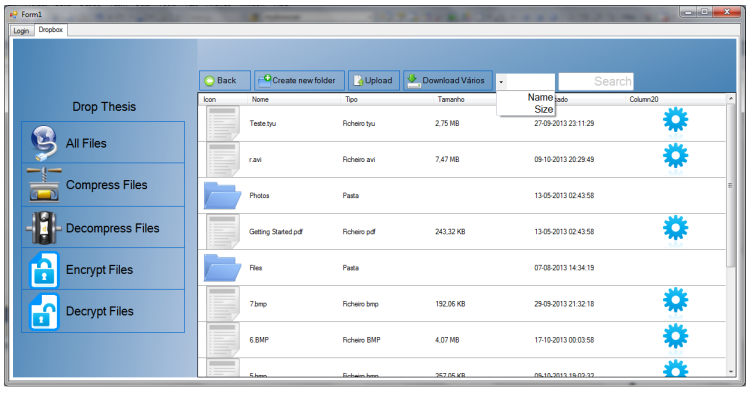

Figure 4.53: Cloud Computing Application: Sort files by name or size.

- Navigate between Multimedia content (Figure 4.54) - using the preview files function is allowed the user to navigate through the previews of the files.

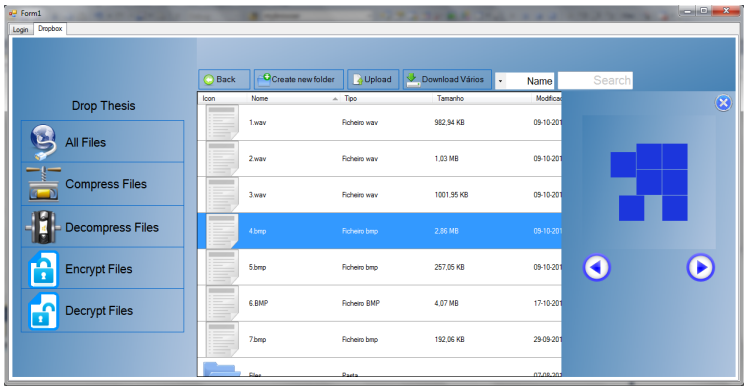

Figure 4.54: Cloud Computing Application: Navigate between Multimedia content.

It is also allowed the user a preview of his file:
- Preview of a file if this is a media content (Figure 4.54).
It is still allowed if the file is a movie, fast forward, rewind time, increase and decrease the volume.

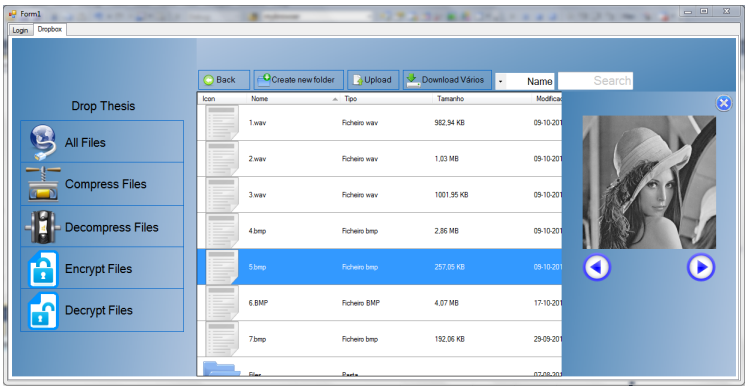

Figure 4.55: Cloud Computing Application: Preview file.

- Compress File (Figure 4.56).

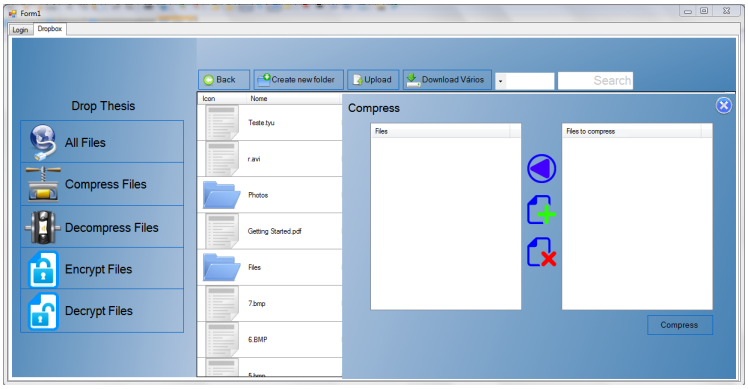

Figure 4.56: Cloud Computing Application: Compress file.

- Decompress File (Figure 4.57).

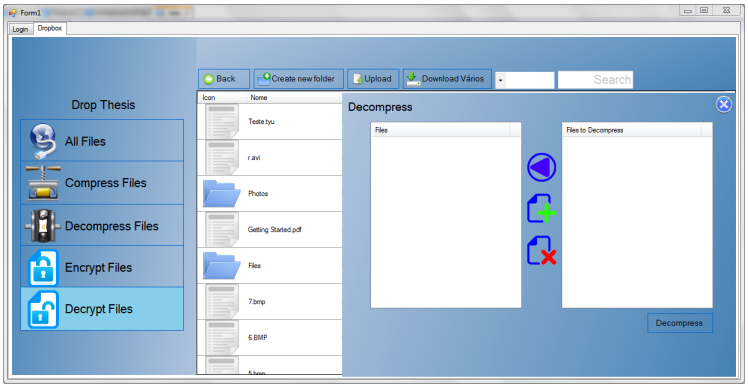

Figure 4.57: Cloud Computing Application: Decompress file.

- Encrypt File (Figure 4.58).

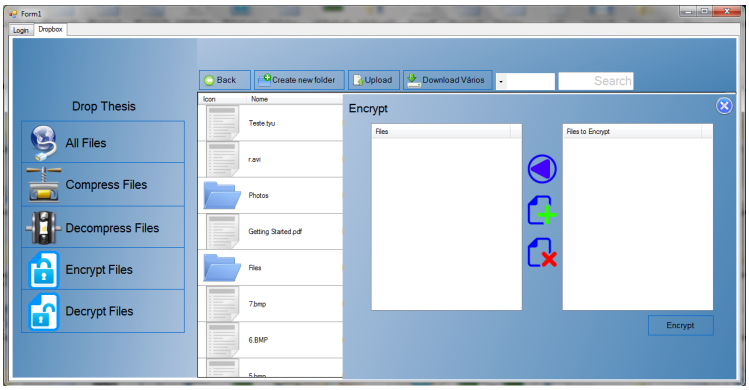

Figure 4.58: Cloud Computing Application: Encrypt file.

- Decrypt File (Figure 4.59).

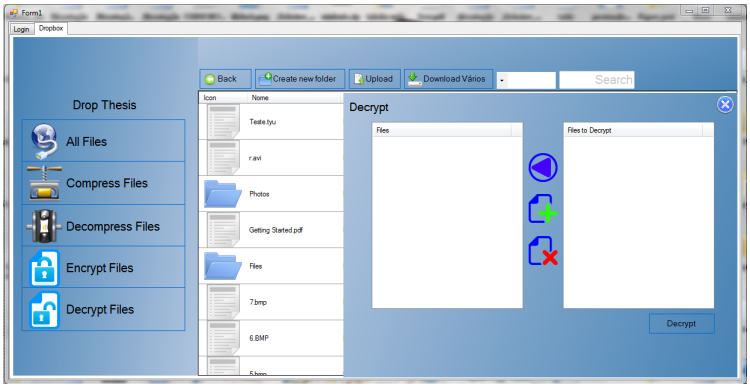

Figure 4.59: Cloud Computing Application: Decrypt file.

4.6 Experimental Validation of the Framework

After the completion of the framework, we proceeded to tests in order to validate experimentally the suitable operation of the framework and to evaluate different parameters associated with the compression, decompression, encryption and decryption of multimedia contents. Table 4.30 shows the results for experiments performed with the implemented prototype of the framework where it is applied compression and then encryption over some data stored in the cloud. As may be seen in this table, the obtained results show that compression followed by encryption may be effective for efficient and secure storage of multimedia content. On the other hand, if we aply first encryption followed by compression, the compression rate is very small (see table 4.31) due to the higher degree of ramdomness introduced by encryption. These results confirm the unsuitability of encryption followed by compression for eficient multimedia storage. Figure 4.60 ilustrates the images corresponding to the files 1.bmp, 2.bmp and 3.bmp used in the experiments reported above.

Table 4.30: Results for experiments with compression followed by encryption.

File	Type	Size	Compression Time	Encryption Time	Decryption Time	Decompression Time	Compress Ratio
1.bmp	Image	2.85MB	16.3286s	0.1062s	0.1631s	21.9829s	70.95%
2.bmp	Image	260KB	5.2409s	0.0294s	0.0659s	4.6615s	32.31%
3.bmp	Image	4.1MB	30.6345s	0.3177s	0.4201s	32.5879s	58.54%
1.wav	Sound	984KB	11.3371s	0.1226s	0.1423s	11.0826s	20.33%
2.wav	Sound	1.03MB	11.8653s	0.1479s	0.1695s	12.1578s	14.57%
3.wav	Sound	0.97MB	10.9208s	0.1361s	0.1581s	40.4947s	49.77%

Table 4.31: Results for experiments with encryption followed by compression.

File	Type	Size	Encryption Time	Compression Time	Decompression Time	Decryption Time	Compress Ratio
1.bmp	Image	2.85MB	0.3675s	28.2787s	26.5426s	0.5254s	7.2%
2.bmp	Image	260KB	0.0470s	5.4144s	6.4687s	0.0529s	8.7%
3.bmp	Image	4.1MB	0.5035s	43.1995s	47.5873s	0.9057s	6.8%
1.wav	Sound	984KB	0.1314s	12.7684s	0.1578s	0.1892s	3.4%
2.wav	Sound	1.03MB	0.1491s	12.3482s	11.1272s	0.1969s	8.1%
3.wav	Sound	0.97MB	0.1237s	12.2627s	13.1015s	0.1849s	6.3%

4.7 Conclusions

The construction of application came true according to the established work plan and objectives have been achieved. The built application allows to perform media data storage, lossless compression and encryption in various cloud environments.

Testing was performed in their various application features and found by analysis of the results tables that the order of operations compression / encryption has implications both in terms of compression rate, as the level of the time required to perform the compression of the file.

Figure 4.60: Images Used in the Tests.

This is due to the increase of the entropy which difficults the file compression process. Therefore compression must be done in the first place instead of encryption for faster efficient solutions.

Chapter 5

Conclusions and Future Work

5.1 Main Conclusions

In this final part of our work, we will not be lengthened. We will make a few closing remarks that although we thus designate, may serve as a starting point for future work.

The Cloud Computing arises with the appearance of so-called Web2.0 and internet available to all, the information available began to grow exponentially, data sets of such size and complexity are difficult to capture, store, search and analyze second traditional methodologies. All this information is stored in powerful centers spread all over the world. To meet this growth, processors and equipment storage and processing of information does not stop increasing in number, capacity and processing speed. Cloud services are presented as a way of sharing, secure storage to keep important data saved, services that the business user or home user is increasingly seen as reliable. But the complexity of these systems does not occur only from the fact that they contain a lot of information, complexity arises also the need to ensure that all information is stored and maintained in recent formats available on the market This indicates that the tools to handle this information will be changed over time and to ensure that the information is not lost, the safety standards are constantly improved.

The literature reviews allowed the organization of knowledge about the problematic Secure and Efficient Transmission and Storage of Multimedia Content in Public Cloud Environments and investigate different opinions of authors that contribute to the development of Cloud Computing.The main objective of this study was to propose, implement and validate a framework, transparent to the user, to efficient and secure data transfer and storage in public cloud environments. Throughout this study and at the end of the implementation and validation of the framework, the objective was achieved.

Throughout our study, we found ourselves with some limitations, primarily in terms of time for implementation that was not allowing scarce other features in the application.

5.2 Directions for Future Work

In future research it would be appropriate to continue the project expanding the potentialities of the framework, such as testing other compression algorithms, implementing new security methods, test the functioning with other users and make comparative tests and expand the scope of the framework implementing more cloud services.

Bibliography

[1] Amazon. Welcome to S3. Available from: http://docs.amazonwebservices.com/AmazonS3/latest/dev/Welcome.html?r=3744 , 2006.

[2] Abadi , D. J. Data Management in the Cloud: Limitations and Opportunities. IEEE Data Engineering Bulletin, 32(1). 2009.

[3] Chen, Y; Paxson, V. and Katz R. H. What's New About Cloud Computing Security? Rel. téc. UCB/EECS-2010-5. EECS Department, University of California, Berkeley, 2010. URL:http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html.What's New About Cloud Computing Security? EECS at UC Berkeley www.eecs.berkeley.edu

[4] Nicolae, B. High Throughput Data-Compression for Cloud Storage. Bogdan University of Rennes 1 IRISA, Rennes, France, 2010.

[5] Goda K., member IEEE, and Kitsuregawa, M., senior member IEEE. The History of Storage Systems. Vol. 100, May 13th, 2012.

[6] Daniel, E. D.; Meem, C. D. and Clark, M. H. Magnetic Recording: The First 100 Years. New York: IEEE Press, 1999.

[7] Engel, F. K. Oberlin Smith and the Invention of Magnetic Sound Recording: An Appreciation on the 150th Anniversary of the Inventor's Birth, 2006. [Online]. Available:http://www.richardhess.com/tape/history/EngelVOberlin_ Smith_2006.pdf.

[8] Eckert-Mauchly Computer Corp., Preliminary Description of the UNIVAC, 1956.

[9] Seagate Technology, ST506 MicroWinchester OEM Manual Preliminary, 1981.

[10] Peek, H. B., BThe emergence of the compact disc, IEEE Commun. Mag., vol. 48, no. 1, pp. 10–17, Jan. 2010.

[11] Gal, E. and Toledo, S. BAlgorithms and data structures for flash memories, ACM Comput. Surv., vol. 37, no. 2, pp. 138–163, 2005.

[12] Smith, R. *Computing in the cloud*. Research-Technology Management, Industrial Research Institute, v. 52, n. 5, pp. 65-68, 2009.

[13] Gillet S, Kapor M. The Self-governing Internet: Coordination by Design, Massachusetts Institute of Technology, USA, 1997.

[14] Aymerich, F., M., et al. *An aproach to a cloud computing network, In: Proceedings of the 1st International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*. Washington, DC (US): IEEE Computer Society, 2008.

[15] Mohamed, A. *A history of Cloud Computing*. ComputerWeekly.com, 2009.

[16] Kaufman, L. M. *Data Security in the World of Cloud Computing*. IEEE Security and Privacy, 2009.

[17] Computerweekly. A history of cloud computing, Internet, cited 2013 maio 23, 2009 Available from: http://www.computerweekly.com/feature/A-history of-cloud-1computing.

[18] IBM, Seeding the Clouds : Key Infrastructure Elements for Cloud Computing Seeding the Clouds : Key Infrastructure Elements for Cloud Computing, Available from: http://resources.idgenterprise.com/original/AST-0001412_ibm_seedingtheclouds.pdf ,2009.

[19] Baca S, Instructor GK. Expert Reference Series of White Papers Cloud Computing : What It Is and What It Can Do for You Cloud Computing : What It Is and. Knowledge Creation Diffusion Utilization [Internet]. 2010;1–6. Available from: http://www.globalknowledge.se/content/files/documents/338677/White_Paper_Cloud_Computing_What_It_Is.pdf.

[20] Raghupathi K. *5 Key Events in the history of Cloud Computing*, Available from: http://cloud.dzone.com/news/5-key-events-history-cloud, 2011

[21] Gomes, C. *Estudo do Paradigma Computação em Nuvem*, Instituto Superior de Engenharia de Lisboa, Área Departamental de Engenharia Eletrónica e Telecomunicações e de Computadores, 2012.

[22] McEvoy, G. V. e B., Schulze. *Using Clouds to Address Grid Limitations*, Proceedings of the 6th international workshop on Middleware for grid computing, ACM, Belgium, 2008.

[23] Staten, J. *Is Cloud Computing Ready for the Enterprise?*, Forrester Research Study, 2008.

[24] Vaquero L. M., Rodero-Merino L., Caceres J., and Lindner M. *A break in the clouds: Towards a cloud definition*, SIGCOMM Computer Communications Review, 39:50 – 55, 2009.

[25] Rimal B. P., Jukan A, Katsaros D, Goeleven Y. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. *Journal of Grid Computing. pp. 3-26.* Available from: http://www.springerlink.com/index/10.1007/s10723-010-9171-y, 2010

[26] Rappa, M. A. *The Utility Business Model and the Future of Computing Services*, IBM Syst. J. 43(1), 2004.

[27] National Institute of Standards and Technology (NIST). *The NIST definition of cloud computing*. Gaithersburg, 2009.

[28] Damiani G. *Cloud Computing: Emerging Technology Executive Briefing* (Part I of II), NIST National Institute of Standards and Technology, Visual Model of Cloud Computing, 2013.

[29] Banerjee, P. Et al. and Alistair Veitch. Everything as a Service: Powering the New Information Economy, Hewlett-Packard Laboratories, IEEE, 2011

[30] Jensen, Meiko et al. *On technical security issues in cloud computing*. In: Proceedings of the IEEE International Conference on Cloud Computing. Washington, DC (US): IEEE Computer Society, p. 109-116, 2009.

[31] Jericho. *Cloud cube model: selecting cloud formations for secure collaboration.* San Francisco, CA (US): Jericho Forum, 2009.

[32] Voas, J.; ZHANG; J. *Cloud Computing: New Wine or Just a New Bottle?* IT Professional, 11(2): 15-17, 2009.

[33] Dikaiakos, M. D.; Pallis, G.; Katsaros, D.; Mehra, P.; Vakali, A. *Cloud Computing - Distributed Internet Computing for IT and Scientific Research*, IEEE,Computing, 2009.

[34] Cloud Security Alliance (CSA). *Security Guidance for Critical Areas of Focus in Cloud Computing* - Version 2.1. Cloud Security Alliance, 2009.

[35] Voorsluys W., Broberg J., Venugopal S., and Buyya R. *Cost of virtual machine live migration in clouds: A performance evaluation, in Proceedings 1st International Conference on Cloud Computing*, Beijing, pp. 254 – 265, 2009.

[36] Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A. *D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; Zaharia, M. Above the Clouds: A Berkeley View of Cloud Computing,* EECS Department, University of California, Berkeley, 2009.

[37] Westerman, G. & Hunter, R. *O Risco de TI: Convertendo ameaças ao negócio em vantagem competitiva*. M.Books, 2009.

[38] ENISA, European Network and Information Security Agency. *Cloud Computing Risk Assessment, 2009*.

[39] Kim, W. Cloud computing: today and tomorrow. *Journal of Object Technology*. Zurich: ETH Zurich, v. 8, n. 1, pp. 65-72, 2009.

[40] Maxey, M. *Cloud Computing Public or Private?* How to Choose Cloud Storage, SysCon.com. http://cloudcomputing.syscon.com/node/707840, 2008.

[41] Rean. *Above the Clouds blog: Surge Computing/Hybrid Computing*, Berkeley, University. http://berkeleyclouds.blogspot.com/2009/05/surgecomputing.html.

[42] Albuquerque, R. and Ribeiro, B. *Segurança no desenvolvimento de Software*, Editora Campus, 2002.

[43] Abd-El-Malek M., Ganger, G. R., Goodson, G. R., Reiter, M. K. e Wylie, J. *Fault-scalable Byzantine fault-tolerant services*. : "Proceedings of the twentieth ACM symposium on Operating systems principles". ACM SOSP '05. Brighton, United Kingdom, pp. 59–74, 2005.

[44] Rezende, D. A. and Abreu, A. F. *Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais*. Atlas, 2000.

[45] Sêmola, M. *Gestão da Segurança da Informação: Uma visão Executiva*, Editora Campus, 2003.

[46] Filho, O. & Neto, H. *Processamento Digital de Imagens,* Rio de Janeiro: Brasport, 1999.

[47] Pescatore, J. (2013). http://www.computerworld.com.pt/2012/05/28/cloud-*computing-exige-seguranca-especifica*/.12 de junho de 2013.

[48] Estrela, J. M. M. *Segurança em redes de computadores*, Master's thesis, Faculdade de Engenharia da Universidade do Porto, 1998.

[49] Soares, J. F. *Interpretação da segurança de sistemas de informação segundo a teoria de ação*. Master's thesis, Universidade do Minho, 2005.

[50] Gonsalves, A. *Computer World*, http://www.computerworld.com.pt/2013/06/11/mit-progride-na-seguranca-para-cloud-computing/, 2013.

[51] Serrão, C. *Gestão de sistemas de informação* (Technical Report, ISCTE/DCTI.), 2009.

[52] Wang, C. and Ren K. *Toward Publicly Auditable Secure Cloud Data Storage Services*. Illinois Institute of Technology Wenjing Lou, Worcester Polytechnic Institute Jin Li, IEEE Network, July/August 2010.

[53] Wang, Q., Wang C., Ren, K., Lou, W., and Li, J. *Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing*. IEEE Transactions on parallel and distributed systems, vol. 22, nº. 5, pp.847-859, may 2011.

[54] Harnik, D., Pinkas, B., Shulman, A. *Side Channels in Cloud Services Deduplication in Cloud Storage*. Copublished by the IEEE Computer and Reliability Societies,pp. 40-47, 2010.

[55] Wang, C., Wang Q., Ren, K., Cao, N., and Lou, W. *Toward Secure and Dependable Storage Services in Cloud Computing*. IEEE Transactions on services computing, vol. 5, no. 2, pp.220-231, April-June 2012.

[56] Zhu Yan, Ahn, G., Hu, H., Yau, S., An Ho G., and Hu, Chang-Jun. *Dynamic Audit Services for Outsourced Storages in Clouds*. IEEE Transactions on services computing, vol. 6, no. 2, pp.227-237, April-june 2013.

[57] Lin, Hsiao-Ying, and Wen-Guey Tzeng. A Secure Erasure *Code-Based Cloud Storage System with Secure Data Forwarding, IEEE Transactions on parallel and distributed systems, vol. 23, no. 6,*pp.995-1002, pp.995-1002,June 2012.

[58] Yang K. and Jia X. *An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing*. IEEE Transactions on parallel and distributed systems, vol. 24, no. 9, pp.1717-1725, september 2013.

[59] Zhang, Y., Ren, S. , Chen, S. , Tan, B. , Lim, S. and Yong, K. DifferCloudStor: Differentiated Quality of Service for Cloud Storage. IEEE Transactions on magnetics, vol. 49, no. 6, pp.2451-2457, June 2013.

[60] Li M., Yu, S., Ren, K., Lou, W. and Y. Hou, T. *Toward Privacy-Assured and Searchable Cloud Data Storage Services*. IEEE Network, July/August 2013

[61] RSA Security Inc, *O papel da segurança na computação em nuvem confiável*, 2009.

[62] Ribeiro N., J. Torres, *Tecnologias de Compressão Multimédia, FCA* - Editora de Informática, Lda, September 2009.

[63] Salomon, D. *Data Compression: The Complete Reference*. Nova Iorque: Springer, 2000.

[64] Bhaskaran, V. & Konstantinides, K. *Image and Vídeo Compression Standards: Algorithms and Architectures*. (2nd. Ed.) Norwel, MA: Kluwer Academic Publishers, 1997.

[65] Buford J. *Multimedia Systems*. Reading, MA: ACM/Addison-Wesley, 1949.

[66] Fluckiger F. Undertanding. *Networked Multimedia: Aplications and Technology*. Prentice Hall, London, 1995.

[67] Shannon C. Prediction and Entropy of Printed English. *Bell System Technical Journal, 1951.*

[68] Li, Z., Drew M. *Fundamentals of Multimedia*. Prentice-Hall, 2004.

[69] Shannon C. A Mathematical Theory of Communication. *Bell System Technical Journal, 1948.*

[70] Huffman, D. *A Method for the Construction of Minimum-Redundancy Codes*. Proc. IRE, 1098-1101, 1952

[71] Pennebaker, W. B., Mitchell, J. L. *JPEG Still Image Data Compression Standard*, Ed. Van Nostrand Reinhold, 1993.

[72] Sanches, I. *Compressão sem Perdas de Projeções de Tomografia Computadorizada usando a Transformada Wavelet,* Dissertação de Mestrado, Departamento de Informática, UFPR, Curitiba, 2001.

[73] Huang, K., Smith, B., Experiments with a Lossless JPEG Codec, June. (1994)."http://www.cs.a>mell.edi^nfo/Projects/zeno/Projects/LJPG.html".

[74] Neto, J.F., Alcocer, P.R.C. *Compressão de Imagens Médicas Utilizando a Técnica JPEG-DPCM,* IV Fórum Nacional de Ciência e Tecnologia em Saúde, pp. 411-412, Curitiba, 1998.

[75] Hammer-Lahav , E. Internet Engineering Task Force (IETF), Ed. Request for Comments: 5849 April 2010 Category: Informational ISSN: 2070-1721, http://tools.ietf.org/html/rfc5849.RFC 5849 - The OAuth 1.0 Protocoltools.ietf.org.

[76] Yahoo, Developer Network, OAuth, http://developer.yahoo.com/oauth/guide/oauth-auth-flow.html.

[77] Fielding, R. Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine, 2000.

[78] Crockford, D. The application/json Media Type for JavaScript Object Notation (JSON). Network Working Group: http://tools.ietf.org/html/rfc4627. 2006.