**CHAPTER 1**

**INTRODUCTION**

* 1. **Overview**

Data is one of the most valuable assets that any company can hold. One of the best ways to store these assets is within the cloud. Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

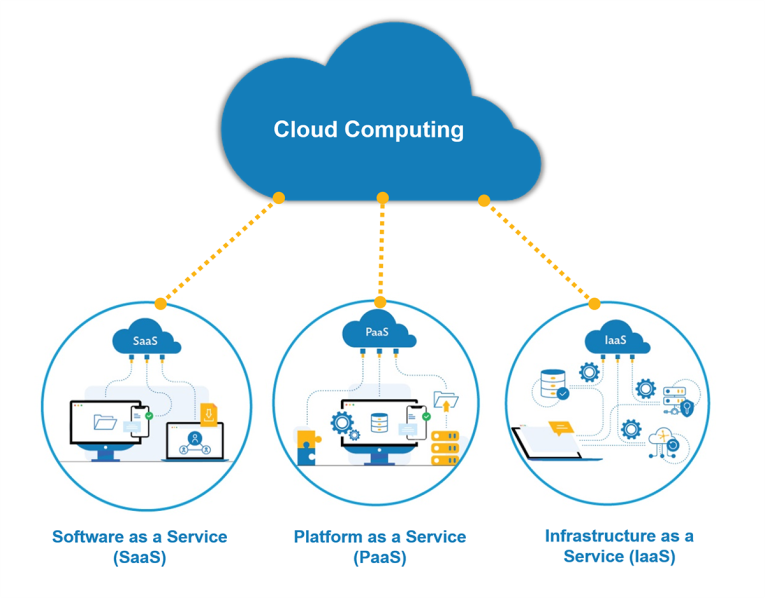


Figure 1.1. Types of Cloud Computing

* + 1. **Types of Cloud Services**

Cloud computing is not a single piece of technology like a microchip or a cell phone. Rather, it's a system primarily comprised of three services: software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS).

* **Software-as-a-service (SaaS):** It involves the licensure of a software application to customers. Licenses are typically provided through a pay-as-you-go model or on-demand. This type of system can be found in Microsoft Office's.
* **Infrastructure-as-a-service (IaaS)**: Involves a method for delivering everything from operating systems to servers and storage through IP-based connectivity as part of an on-demand service. Clients can avoid the need to purchase software or servers and instead procure these resources in an outsourced, on-demand service. Popular examples of the IaaS system include IBM Cloud and Microsoft Azure.
* **Platform-as-a-service (PaaS)**: It is considered the most complex of the three layers of cloud-based computing. PaaS shares some similarities with SaaS, the primary difference being that instead of delivering software online; it is a platform for creating software that is delivered via the Internet. This model includes platforms like Salesforce.com and Heroku.
  1. **Problems Identified**

Businesses are rapidly adopting cloud platforms to improve efficiency and agility, and the global pandemic has only accelerated this move. It’s no surprise that almost 70% of organizations currently using cloud services plan to increase their cloud spending.

* + 1. **Cloud Disaster**

In today’s competitive environment, businesses cannot afford to suffer downtime, whether from hardware or software failure, cyber-attacks, or natural disasters. Businesses must be constantly online to meet the burgeoning demands of a global, 24×7 economy.



Figure 1.2. Cloud Disaster

**Three main categories of disaster can affect businesses:**

* **Natural disasters**: Natural disasters such as floods or earthquakes are rarer but not infrequent. If a disaster strikes an area that contains a server that hosts the cloud service you’re using, this could disrupt services and require disaster recovery operations.
* **Technical disasters**: Perhaps the most obvious of the three, technical disasters encompass anything that could go wrong with cloud technology. This could include power failures or a loss of network connectivity.
* **Human disasters**: Human failures are a common occurrence and are usually accidents that happen whilst using cloud services. These could include inadvertent misconfiguration or even malicious third-party access to the cloud service.
  + 1. **Cloud Disaster Recovery**

Cloud disaster recovery is an effective way to build an organization’s resilience against disasters. It helps protect important workloads regardless of whether they are stored on-premises, in the cloud, or in hybrid or multi-cloud environments. A robust cloud DR solution helps businesses cope with cyber threats and other disasters more efficiently, and [minimizes downtime](https://www.unitrends.com/blog/downtime-causes-costs-and-how-to-minimize-it) and the costs associated with it. Fog Computing ensures critical files, systems and applications are secure and available, thereby enabling business continuity.

* 1. **Fog Computing**

Fog computing is a decentralized computing infrastructure or process in which computing resources are located between the data source and the cloud or any other data center.

* + 1. **Fog computing vs edge computing**

For every new technological concept, standards are created, and they exist to provide users with regulations or directions when making use of these concepts. In the case of the edge and fog computing, while edge computing refers to bringing compute closer to data sources, fog computing is a standard that defines its operation and application in diverse scenarios.

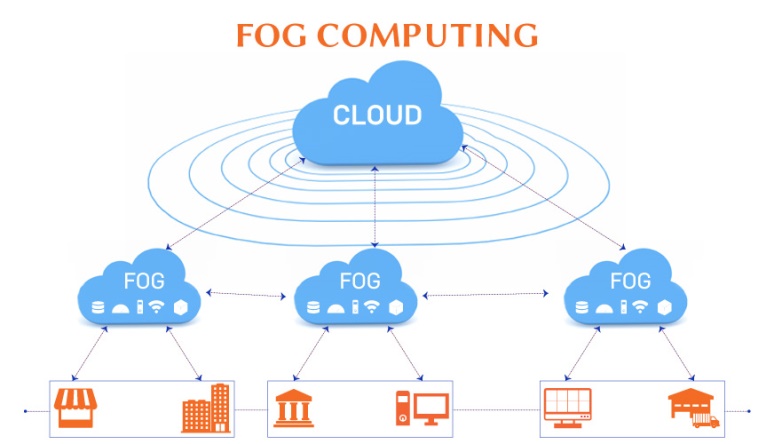


Figure 1.3. Fog Computing

Fog computing was coined by Cisco, and it enables uniformity when applying edge computing across diverse industrial niches or activities. This makes them comparable to two sides of a coin, as they function together to reduce processing latency by bringing compute closer to data sources.

* + 1. **Fog computing: verticals and use cases**

Verticals range from transportation and logistics (the latter in the logistics 4.0 scope), smart buildings and cities, IoT in healthcare and utilities/energy to agriculture, oil and gas, mining and also residential and consumer verticals. And of course, smart manufacturing, the eternal number one industry from an IoT spending perspective.

* 1. **Motivation of the Project**

Disaster recovery can be used to manage entire network backup, including applications, email, data and network configuration. If an IT failure strikes, data and applications can be restored at any time from The Cloud. Business’ IT workloads can be disrupted at any time, so it is always important to have a recovery plan prepared. One advantage of cloud disaster recovery is that it can be used to back up and restore data that run on-premises, in addition to those hosted in The Cloud (offsite server).

* 1. **Scope of the Project**
* Even a minor outage can put at a competitive disadvantage.
* The scope of the project is to ready with a business continuity plan that includes cloud backup and disaster recovery for critical IT systems — without the expense of secondary infrastructure.
* Bring the own backup and disaster-recovery tools to the cloud at the fog node.
  1. **Objective of the Project**

The objective of the project is to propose HAEdge and VirtualBox, a new backup solution to tackle the problem of data security and reliability. The solution is based on Fog computing paradigms.

* To Protect Data Effortlessly with VirtualBox
* To back up the data in a compressed manner and store it in VirtualBox with minimal storage space.
* Tomitigate the risk of downtime and protect sensitive data.
* To restore service once a disaster has been identified.

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **Fog Computing Advancement**

**By Rosilah hassan** (2019): This article is to conduct a complete and fundamental survey to study the reality of Fog Computing technology on our real-life and its impact by understanding its core concept. We also define FC based on previous studies and clarify the difference between cloud computing and Fog Computing concerning other related technologies. Moreover, the architecture of FC was tackled in this review article based on FC literature. We demonstrate that FC has a flexible environment to be adjusted based on the users’ customized requests. The limitations of cloud computing enabled Fog Computing to be a significant technology that serves as a broker for iot to facilitate the practical deployments.

**2.2 Data Consistency in Multi-Cloud Storage Systems With Passive Servers And Non-Communicating Clients**

By Naram mhaisen (2020): This article discussed multi-cloud storage systems and their significant advantages in establishing trust in the cloud. The paper focuses on addressing the concurrent data access issue and reasons why it is especially challenging in such systems compared to conventional storage systems. Various previous solutions to this issue were discussed. The paper offered a useful formal definition of data consistency and data conflicts in a multi-cloud storage system and proposed a novel method to detect data conflicts and maintain data consistency.

# 2.3 Multi-Replica and Multi-Cloud Data Public Audit Scheme Based on Blockchain

By [Ting Li](https://ieeexplore.ieee.org/author/37086852652) (2020): This article proposes a multi-replica and multi-cloud data public audit scheme, which not only supports the modification, insertion and deletion of cloud multi-replica data, but also can track the Data of malicious users. In addition, block chain technology is introduced to restrict the behaviour of third-party auditors. The analysis results show that our scheme satisfies the enforceability of tag and the robustness of audit and can resist malicious auditors. Compared with the similar scheme, our scheme has higher performance in communication and computation overhead.

**2.4 Fog Computing: A Comprehensive Architectural Survey**

By [Alberto Leon-Garcia](https://ieeexplore.ieee.org/author/38270617400) (2020): This article, a comparative study of competing technologies including cloud computing, mobile computing, edge computing and fog computing was carried out. Also, taxonomy of fog computing research was developed that addresses various subject areas (system, application, software, security, resource management and networking) as well as research aspects, namely architecture, algorithms and technologies. A comprehensive survey of various architectural perspectives in fog computing is provided. These architectural perspectives complementarily elaborate on physical as well as logical components and modules in a fog computing environment and their respective roles and functionalities. Some architecture has a network wide view and some focus on node-level architecture. Related algorithms and technologies need to be developed in a consistent and robust manner to realize the full advantage of fog computing, some of which has already taken place.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 Existing System**

* **Microsoft Azure**

Azure offers physical and virtual support for Linux, Windows, VMware virtual machines, Windows Server, and System Center management tools. It is a solid choice for companies who are interested in protecting their critical workloads via Hyper-V or even VMware, especially companies who are already on the Microsoft software stack.

* **Zetta Backup and Recovery**

Zetta offers cost-effective DRaaS with user-friendly setup onto Zetta’s data center.

* **Amazon Web Services (AWS)**

AWS offers DR for various databases and enterprise applications such as MySQL, SQL Server, and even SAP. AWS uses their Cloud Endure Disaster Recovery service to continually replicate the OS, databases, applications, and files into your Region of choice.

* **Zerto**

Zerto provides a user-friendly interface and flexible solution that can convert between VMware, Hyper-V, and even Amazon Web Services (AWS).

* **Carbonite Cloud Backup**

Carbonite offers an affordable solution with unlimited server licenses. It also supports Windows and Mac OS X operating systems.

* **Triviback**

It is a chunking-based backup system that minimizes the storage needs using the sec-cs data structure for deduplication of flat contents.

* **Trusty Drive**

It is a document storage system on multiple cloud providers. It tries to preserve user anonymity and document anonymity.

* **ExpanStor**

It is another multi-cloud storage system with dynamic data distribution. It applies a Client-Server architecture instead of a pure Client-Based implementation.

**3.2 Disadvantages**

* Speedy recovery is crucial to mitigate downtime and high latency could mean added downtime.
* Data redundancy allows your organization to store its data in two separate places.
* The upfront capital expenditures associated with building a second data repository, however, are numerous. There are warehouse costs, maintenance costs, hardware costs, and transportation costs if data is stored on disks and tapes.
* Hire IT technical staff to develop, test, and execute your DR strategy.
* Back up your systems and data and maintain these backups in an on-site data center, physical or virtual machine (VM), or disk or tape stored on-site.
* Invest in hardware, software, maintenance, and support for hardware and software.
* Disaster recovery used to be complex and expensive.
* Traditional backup services have proven to be less than reliable when it comes to data protection with regular disruptions and loss of data.

**3.3 Proposed System**

The proposed System introduces HAEdge, a new data backup system based on Fog Computing. This system utilizes the advantages of Fog-Cloud storage to ensure users' data protection and reliability and, at the same time, overcomes the problems of multi-Cloud using the Fog Computing paradigm. System users can easily and securely backup, restore, and modify their data without caring about the sophisticated operations to protect and secure the data on multi-Cloud storage. Proposed VirtualBox to provide an easy-to-use, highly secure, and reliable backup system using state-of-the-art Cloud and encryption techniques.

* **VirtualBox:** It is a personal Fog node. It is owned and controlled by the end-user, similar to the rest of his personal devices like smartphones and laptops. VirtualBox can be considered a Private Fog device. This is, somehow, similar to the Private Cloud concept, where the organizations deploy and manage their own cloud infrastructure. VirtualBox is located within the network architecture. Being a personal device, the VirtualBox enables many applications to utilize the Fog Computing paradigm. VirtualBox benefits from this advantage to provide a better backup experience and a unique system that outperforms the existing systems.

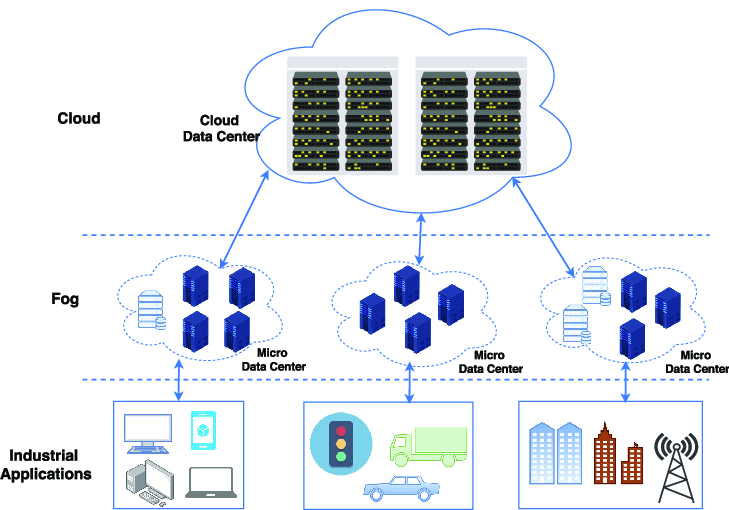


Figure 3.1 Fog Drive

* **HAEdge:** HAEdge provides a unique data backup system architecture. This uniqueness is derived from the combination of Fog Networking and Multi-Cloud advantages. Whereas multi-cloud methods are used to provide the most reliable and secure storage environment, Fog Computing is used to provide better throughput and lower latency for the backup process. The enabler of this unique architecture is the VirtualBox fog device.

The HAEdge provider manages the entire backup process, providing the client with a user-friendly dashboard to manage their data in the cloud and VirtualBox. Fog-based data protection services are of three types:

* Backup as a Service (BaaS)
* Recovery as a Service (RaaS)
* Disaster Backup Recovery as a service (DBRaaS)

Using BaaS, Cloud Service Providers (CSPs) backup and restore the contracted data. RaaS is an extension of BaaS, where applications are also restored from the cloud in addition to the data. DRaaS is much more extensive than RaaS and includes cloud-based failover, in addition to backup and recovery services. DRaaS adoption is essential for organizations requiring 100% availability of their critical applications.

**OpenPGP**

OpenPGP (Pretty Good Privacy (PGP)) Scheme based Encrypted backups allow your personal control over your data by locking it up with passwords meaning only your authorized users can access it. After encrypting files with PGP, can place them on the network or transfer them to a remote server. This process can be automated from end to end. For example, a project workflow could be defined to automatically retrieve records from a database, create a CSV file from those records, encrypt the file, and send it to your trading partner's SFTP server, all without manual interference from you or your team.

**Open PGP Key Management**

A comprehensive Key Manager is provided PGP public and private keys. This Key Manager can be used to create keys, change keys, view keys and import keys. These keys can be utilized within Cloud – Fog Data for automating PGP encryption and decryption within your organization. This Key Manager can also be used to export public keys for sharing with your trading partners.

**Encryption**

Encryption is performed through the use of a key pair. The public key may be published or sent to the recipient. The private key is known only to the recipient, who will decrypt the key. The public key is used to encrypt the message or the file and the private key to decrypt it.

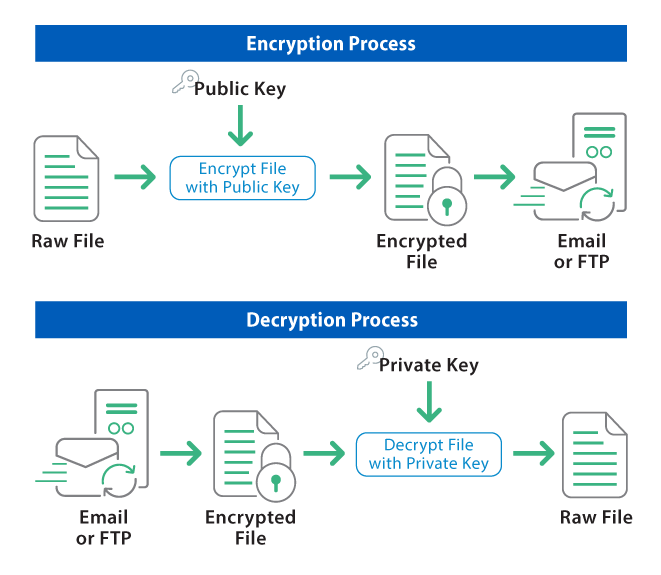


Figure 3.2. OpenPGP

**Standards Support for Open PGP Encryption**

The Open PGP standard is a non-proprietary and industry-accepted protocol that defines the standard format for encrypted messages, signatures and keys.

**Asymmetric Encryption Algorithms**

* Diffie-Hellman
* DSA
* RSA

The key sizes supported are 512, 1024, 2048 and 4096 bits.

**Ciphers**

* AES-128
* AES-192
* AES-256 (default)
* Blowfish
* CAST5
* DES
* IDEA
* Triple DES(DESede)
* Twofish

The default symmetric algorithm is AES-256, which can be changed by the user.

**Hash Algorithm**

* MD2
* MD5
* RIPEMD-160
* SHA1 (default)
* SHA-256
* SHA-384
* SHA-512

The default hash algorithm is SHA1, which can be changed by the user.

**Compression Algorithm**

Compression is basically converting a message of n bits to m bits (n > m) using a compression algorithm. Compression helps the email service providers to increase their productivity as the storage overhead, processing and labour spent on the maintenance of their servers is reduced.

PGP also optionally performs compression on files using compression methods that include:

* bzip2
* zip (most common)
* zlib

Compression service in PGP is provided using ZIP Algorithm. The Compression is included in combined Authentication & Confidentiality of PGP as follows:

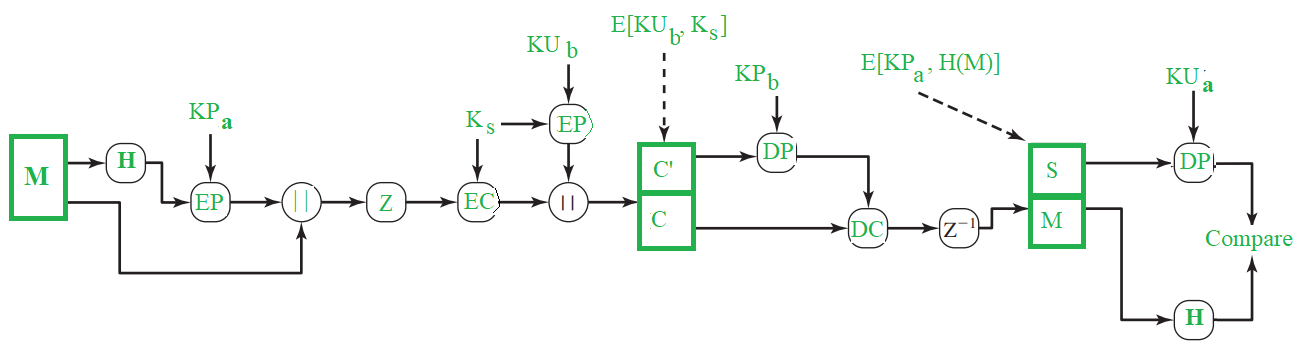


Figure 3.3. PDP Compressor

In the figure,

Z– Compression function

Z-1– Decompression function

In PGP, message is compressed only after the application of Signature. The compressed data is decompressed at the receiver’s end to obtain the original message and the signature. Then, we can extract the hash value from the signature and then we can directly determine the authenticity right away by simply calculating the message’s hash value and comparing it with the value obtained from the signature.

If compression was done before the signature was applied, which means that the hash value in the signature is not that of the original message but is of the compressed message. Thus, for checking the authenticity at the receiver’s end, we would have to either maintain the compressed message or to recompress the message. Because of the PGP algorithm we would have another problem at the time of recompression, as the algorithm may produce a variety of results for a input, based on it’s implementation and thus we may have inconsistency in the hash values obtained from them. Hence, there would be inconsistency in Authenticity verification.

Even though we get different outputs of compressed message with different implementations of algorithm, same output is produced for decompression when compressed message from any implementation is given as input. So, in order to achieve a consistent authenticity procedure, we would have to implement compression with a single version. Thus, for a safer use, PGP does the compression after the signature has been applied.

Normally, when both the message and signature are compressed, they are encrypted. It is to provide that extra layer of security from cryptoanalysis as there would be less redundancy than the plaintext.

**3.4 Advantages**

* Mitigating risk, such as snapshots or mirror copy failure & corruption, human error, hardware failure or poorly documented communication.
* Minimizing downtime: keeping your business online!
* Insuring against natural disasters by having geographic diversity: storing copies of your data in local drive.
* Having the ability to withstand human error or physical impact on a production environment.
* Limit the effect on business or downstream customers.
* Automated backup of critical data and systems.
* Minimal human interaction needed for quick disaster recovery.
* Flexibility to restore the entire infrastructure or just a single application.
* Choose the type, source, target and location of your data
* Determine the best route for continuity – full recovery or repopulating data
* Understand current status with built-in alerting, monitoring and reporting
* The ability to automatically backup critical systems and data.
* The ability to quickly recover from a disaster, with minimal user interaction.
* Flexible recovery options, such as restoring a single application or the whole infrastructure.
* Reliability, flexibility and high performance
* Effective addressing of mobility and portability challenges
* Consumption of much lesser operational resources
* Rapid and immediate recovery of data in the event of a disaster
* Quick restoration of normal operations
* Significant savings in terms of software licenses and hardware

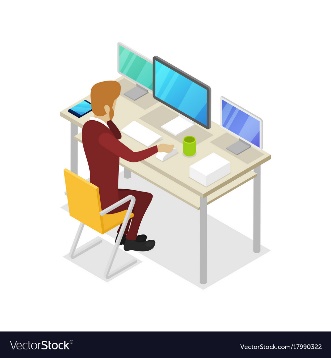
**CHAPTER 4**

**PROJECT DETAILS**

**4.1 Project Description**

The project is the development of cloud-based application, combining both cloud-based and local components. This application relies on remote servers, accessible through web browsers with a constant internet connection. Cloud storage servers, facilitated by third-party cloud service providers, offer virtual storage facilities accessible via the internet. These servers, located in remote data centers, ensure continuous operation and maintenance, safeguarding files from potential damage. Data stored on cloud services is readily available to authorized users, facilitated through application programming interfaces (APIs). These APIs enable seamless communication between the cloud application and third-party data sources or storage services. The use of APIs contributes to the efficiency and predictability of the application, streamlining development processes. Within the cloud storage framework, three distinct types are highlighted: object storage, file storage, and block storage. Object storage, exemplified by services like Amazon Simple Storage Service (S3), offers scalability and metadata characteristics, ideal for modern applications and analytics computations. File storage, as supported by solutions like Amazon Elastic File System (EFS), caters to applications requiring shared files and a file system. Block storage, represented by services like Amazon Elastic Block Store (EBS), is tailored for high-performance workloads, often necessary for enterprise applications such as databases or ERP systems. The project also emphasizes the significance of home directories in cloud workflows. The utilization of home directories allows businesses to extend access to users, leveraging the scalability and cost benefits of the cloud. Cloud file storage solutions adhere to required file system semantics and standard permissions models, facilitating the seamless transition of applications to the cloud. The HAEdge App Layer is a vital component of the project, providing a Windows-based, free, and open-source computer imaging solution. It integrates open-source tools through a PHP-based web interface, supporting tasks such as changing PC hostnames, restarting computers, and managing imports into the VirtualBox database and Cloud Storage Server. The application layer plays a critical role in backup, recovery, and cloud data migration processes. Backup and recovery are essential aspects of data protection and accessibility. The project addresses challenges related to increasing capacity requirements, advocating for the flexibility of storing database backups in the cloud for temporary protection during updates or for development and test purposes. Cloud data migration, while presenting compelling benefits, acknowledges the concerns of traditional IT functional owners, including those related to storage, backup, networking, security, and compliance. VirtualBox, within the project’s framework, serves as an archive storage solution. Cost-effective and suitable for data storage of at least 365 days, VirtualBox is best implemented on a dedicated server or any available spare local machine, contributing to the project’s overall efficiency. Data security and compression are integral components of the project, employing the OpenPGP Protocol to secure data in transit. The encryption and decryption processes involve the use of public and private PGP keys, ensuring the security of data communication. Compression further optimizes the efficiency of data transmission, reducing plaintext size and enhancing overall security. The cloud end user segment focuses on individuals or group members relying on cloud server storage for data maintenance and computation. The project distinguishes between data owners and data users. Data owners are responsible for owning and saving encrypted data blocks at the Cloud Service Provider (CSP), constructing dictionaries for data blocks, and encrypting document collections and index trees.

**4.2 System Architecture – VirtualBox Data Backup**



Login

Upload Data

Encrypt Data

Generate ISO File

Compress Data

Received Data

Backup Data

Update VB

Store Data

HAedge - VB

Open PGP

HAEdge Backup

Data Owner

Cloud Front End Interface UI

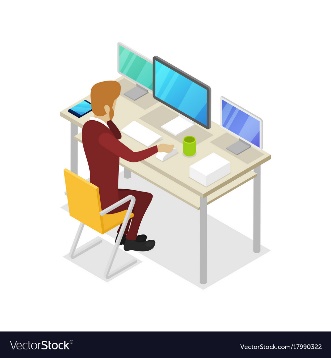
Cloud Backend Server



VB Backend Server

**Fig 4.1 VirtualBox Data Backup**

**4.2.1 System Architecture – HAEdge Data Recovery**



Forward Req toVB

Server Status

Received Req

Check Ser.Avai

Decompress ISO

Request Data

Check SStorage

HAEdge – Virtual Box

Response Data

Request Data

HAEdgeWeb Interface

Retrieve Req.Data

Response Data

Decrypt Data

Data Owner

Data User



VB Backend Server

Cloud Backend Server



Temp.Cloud Server



**Fig 4.2 HAEdge Data Recovery**

**4.3 SYSTEM FLOW**

1. **Detection of Disaster:**

* Continuous monitoring systems detect anomalies or disruptions in the cloud storage server, triggering a disaster response.

1. **HAEdge App Layer Activation:**

* The HAEdge App Layer, designed for computer imaging and disaster recovery, is activated in response to disaster detection.

1. **VirtualBox Archive Retrieval:**

* VirtualBox, serving as an archive storage solution, is accessed to retrieve critical data and configurations.
* Data stored on VirtualBox includes essential system images, configurations, and other necessary information.

1. **Backup Data Restoration:**

* If applicable, the system initiates the restoration of data from backup mechanisms facilitated by HAEdge.
* This may involve copying system images, configurations, and user data from the backup repository.

1. **Parallel Operation with Redundant Systems:**

* Redundant systems provided by HAEdge may be activated to operate in parallel with the affected cloud storage server.
* This ensures continuous data availability and service operation during the recovery process.

1. **Temporary Data Access:**

* Users may be granted temporary access to critical data and services through alternative channels.
* HAEdge and VirtualBox work together to provide access to necessary information during the recovery phase.

1. **Communication with Users:**

* The system communicates real-time updates to users, providing information on the status of the disaster recovery process.
* Users are informed about the expected duration of downtime and any temporary changes in access or functionality.

1. **Continuous Monitoring and Evaluation:**

* Continuous monitoring by HAEdge and VirtualBox ensures the progress and effectiveness of the recovery process.
* Any issues or deviations from the recovery plan are identified and addressed promptly.

1. **Data Verification and Integrity Checks:**

* The restored data from VirtualBox and HAEdge backups undergo verification and integrity checks to ensure accuracy.
* Data consistency is validated to maintain the integrity of critical system images and configurations.

1. **Gradual Resumption of Normal Operations:**

* Once the restoration is successful, the system, with the support of HAEdge and VirtualBox, gradually resumes normal operations.
* Services are restored, and the system transitions from temporary access to full operational status.

1. **Post-Recovery Analysis and Documentation:**

* The system conducts a post-disaster analysis to identify the root cause and evaluate the effectiveness of the recovery plan.
* Documentation of the incident, recovery measures, and lessons learned is crucial for continuous improvement.

1. **Communication of Resolution:**

* Users are informed of the resolution, signaling the restoration of normal operations.
* Any temporary measures implemented during the recovery phase are communicated to users.

1. **Update of Disaster Recovery Plan:**

* Based on insights gained from the disaster, the system updates the disaster recovery plan.
* This includes improvements to enhance the resilience of both HAEdge and VirtualBox systems against future disasters.

**4.4 Modules Description**

1. **Cloud Service Provider Dashboard**

In this module we develop a cloud application, or cloud app, it is a software program where cloud-based and local components work together. This model relies on remote servers for processing logic that is accessed through a web browser with a continual internet connection. Cloud application servers typically are located in a remote data centre operated by a third-party cloud services Infrastructure provider. Cloud-based application tasks may encompass email, file storage and sharing. Third-party data sources and storage services can be accessed with an application programming interface (API). Cloud applications can be kept smaller by using APIs to hand data to applications or API-based back-end services for processing or analytics computations, with the results handed back to the cloud application. Vetted APIs impose passive consistency that can speed development and yield predictable results. Data stored on cloud services is instantly available to authorized users.

**1.1 Cloud Storage Server**

Cloud storage servers are virtual storage facilities provided by cloud service providers that help to store and access multiple files without the requirement of any direct physical device. Web storage server can be accessed via the internet. File Cloud offers the cloud storage in affordable cost and without any downtime. The cloud storage servers continuously run with the help of these data centres and are maintained by the cloud service providers. Data centres secure your files from any kind of damage and make those files available whenever you want to access it via the internet. Applications access cloud storage through traditional storage protocols or directly via an API. Many vendors offer complementary services designed to help collect, manage, secure and analyse data at massive scale. There are three types of cloud data storage: object storage, file storage, and block storage. Each offers their own advantages and have their own use cases:

**Object Storage** - Applications developed in the cloud often take advantage of object storage's vast scalability and metadata characteristics. Object storage solutions like Amazon Simple Storage Service (S3) are ideal for building modern applications from scratch that require scale and flexibility, and can also be used to import existing data stores for analytics, backup, or archive.

**File Storage** - Some applications need to access shared files and require a file system. This type of storage is often supported with a Network Attached Storage (NAS) server. File storage solutions like Amazon Elastic File System (EFS) are ideal for use cases like large content repositories, development environments, media stores, or user home directories.

**Block Storage** - Other enterprise applications like databases or ERP systems often require dedicated, low latency storage for each host. This is analogous to direct-attached storage (DAS) or a Storage Area Network (SAN). Block-based cloud storage solutions like Amazon Elastic Block Store (EBS) are provisioned with each virtual server and offer the ultra-low latency required for high performance workloads.

* 1. **Home directories**

The use of home directories for storing files only accessible by specific users and groups is useful for many cloud workflows. Businesses that are looking to take advantage of the scalability and cost benefits of the cloud are extending access to home directories for many of their users. Since cloud file storage solutions adhere to required file system semantics and standard permissions models, customers can easily lift-and-shift applications to the cloud that need this capability.

**Cloud Service Provider**

**Cloud Storage Server**



**Data Owner**

**Fig 4.3 Home directories**

1. **HAEdge App Layer**

HAEdge is a windows-based, free and open-source computer imaging solution for Windows XP, Vista and 7 that ties together a few open-source tools with a PHP-based web interface. HAEdge also supports putting an image that came from a computer with an 80GB partition onto a machine with a 40GB hard drive as long as the data is less than 40GB. Fog also includes a Web Dashboard service that is used to change the hostname of the PC, restart the computer if a task is created for it, and auto import hosts into the VirtualBox database and Cloud Storage Server.

* **Backup and Recovery**

Backup and recovery is a critical part of ensuring data is protected and accessible, but keeping up with increasing capacity requirements can be a constant challenge. Backing up data using existing mechanisms, software, and semantics can create an isolated recovery scenario with little locational flexibility for recovery. Many businesses want to take advantage of the flexibility of storing database backups in the cloud either for temporary protection during updates or for development and test. Since cloud file storage solutions present a standard file system that can be easily mounted from database servers, they can be an ideal platform to create portable database backups using native application tools or enterprise backup applications.

* **Cloud Data Migration**

The availability, durability, and cost benefits of cloud storage can be very compelling to business owners, but traditional IT functional owners like storage, backup, networking, security, and compliance administrators may have concerns around the realities of transferring large amounts of data to the cloud.

* 1. **VirtualBox**

[Archive Storage:](https://cloud.google.com/storage/docs/storage-classes#archive) Lowest cost. Good for data that can be stored for at least 365 days, including regulatory archives. VirtualBox is best implemented on a dedicated server, any spare machine that company have locally. VirtualBox provides a feature-rich environment for creating and managing virtual machines (VMs), supporting key features like snapshotting for easy rollback and multi-platform compatibility. Creating VMs in VirtualBox is intuitive, thanks to its wizard-driven approach and the inclusion of Guest Additions – additional drivers and utilities enhancing guest operating system performance and integration. The software leverages hardware virtualization capabilities, such as Intel VT-x and AMD-V, to improve overall performance. Networking capabilities are flexible, offering modes like NAT, Bridged, and Host-Only, alongside port forwarding options for tailored network configurations. VirtualBox excels in storage management, supporting various disk image formats and dynamic allocation to optimize storage efficiency. Users can monitor resource usage in real-time, with the ability to set execution caps on CPU usage. The software is extensible, supporting extension packs for additional functionalities and APIs for automation and integration. Top of Form

**Cloud Service Provider**

**Cloud Storage Server**



**Data Owner**

**HAEdge**

**VirtualBox**

**Fig 4.4 HAEdge App Layer**

1. **Data Security and Compression**

In this module we apply OpenPGP Protocol to encrypt and compress data. OpenPGP is an open standard that aims to secure data in transit by enabling end-to-end encrypted communication, widely used to secure cloud data and e-mail services and many other types of applications. OpenPGP currently supports modern cryptography and is constantly vetted by renowned security experts.

**3.1 Encryption and Decryption**

Data management is a central component of OpenPGP. In simple terms, your Data is divided into a pair of public and private PGP keys. The private key must remain a secret and is stored in your device’s keyring. The public key can be shared with your correspondents who can send encrypted files to you that can only be decrypted using your private key. In this setup, unfortunately, an attacker who gains control over your private PGP keys can silently impersonate you, accessing your encrypted traffic and creating valid signatures. While you want to secure these private keys and monitor their usage, applications still need to access them for certain PGP operations.

**Cloud Service Provider**

**Cloud Storage Server**



**Data Owner**

**HAEdge**

**VirtualBox**

**OpenPGP-Encryption**

**Fig 4.5 Data Security and Compression**

**3.2 Zip Engine**

Plaintext is large and takes up an unnecessary amount of modem transmission time and disc space. A PGP program will compress the user’s plaintext to make the entire process more efficient. This also reduces patterns found in plaintext, making it harder for hackers to decipher. Significantly Faster ZIP engine – Our advanced multi-core ZIP/ZIPX engine has been optimized for maximum speed, now up to 30-50% faster than WinZip’s multi-core engine (and much faster than Secure Zip’s and WinRAR’s ZIP engines), while providing similar compressing strength.

**Cloud Service Provider**

**Cloud Storage Server**



**Data Owner**

**HAEdge**

**VirtualBox**

**OpenPGP-Encryption**

**OpenPGP-Compression**

**Fig 4.6 Zip Engine**

1. **Cloud End User**

A Cloud user is an individual user or a group member who outsources large data files on CS. These users completely rely on cloud server storage for data maintenance and computation by Cloud Server Provider (CSP). It has vast storage space and computation infrastructure to maintain and satisfy the cloud user’s needs.

* 1. **Data Owner**

Data holder who owns data and saves its data that consists of multiple blocks at CSP. It is possible that a number of eligible data holders share the same encrypted data blocks in the CSP. In particular, the data holder that first uploads the data blocks to the CSP is denoted as a data owner with regard to the same blocks. Data owner needs to construct a dictionary W, which is composed of distinct keywords extracted from document collection D before outsourcing so that the data availability can be maintained while protecting data privacy. And then, with the dictionary and document collection, an unencrypted index tree can be constructed. Finally, the data owner encrypts the document collection and index tree and outsources encrypted form of them to the cloud server

* 1. **Data User**

Data user can obtain the authorization of accessing a particular document from the data owner. In light of search control mechanisms, the data user can generate a trapdoor T with t query keywords and k encrypted documents will be returned after the trapdoor is uploaded to the cloud server. Finally, with the share secret key, the data user can decrypt documents. The data user obtains a set of attributes as well as corresponding decryption keys from authorities. She/He can verify whether they are legal and available or not. The data user can freely obtain any encrypted data from CSP, and decrypt the ciphertext if and only if her/his attributes satisfy the access policy. Also, data users with insufficient resources can outsource decryption to CSP.

1. **Performance Evaluation**

Performance evaluation of the system to ensure its efficiency and responsiveness. The evaluation focuses on key metrics that directly impact user experience and system reliability. Here's a detailed description of the performance evaluation criteria:

**Time Taken to Retrieve and Restore User Data to VirtualBox**

The primary goal is to assess the system's ability to swiftly recover user data stored in the cloud during a disaster scenario.

* Measure the time elapsed from the initiation of the disaster recovery process to the successful restoration of user data to VirtualBox.
* Consider factors such as the volume of data, network speed, and the efficiency of backup mechanisms in place.
* Evaluate the performance of VirtualBox in handling the data recovery process and its responsiveness to disaster scenarios.

**Time Taken to Compress and Decompress Data:**

This criterion assesses the efficiency of data compression and decompression processes, crucial for optimizing storage space and transmission times.

* Measure the time required to compress a given dataset, considering factors like dataset size, compression algorithms used, and system resources allocated.
* Similarly, measure the time needed to decompress the compressed dataset, evaluating the effectiveness of decompression algorithms and system efficiency.
* Analyze the impact of compression and decompression times on overall system performance, especially in scenarios where data transfer speed is crucial.

**Tasks Success Rates:**

These metric gauges the system's reliability and success in executing various tasks, contributing to user satisfaction and data integrity.

* Define a set of representative tasks that the system routinely performs, such as data backups, cloud storage access, and disaster recovery initiation.
* Execute these tasks under different conditions and scenarios to mimic real-world usage.
* Measure the success rates of these tasks, accounting for successful completions, errors, or any deviations from expected outcomes.
* Analyze the reasons behind task failures, if any, and identify areas for improvement or optimization in the system.

**CHAPTER 5**

**SYSYTEM SPECIFICATION**

**5.1 HARDWARE REQUIREMENTS**

* **Server/Computing Power:** Multi-core processors: Intel Xeon, AMD Ryzen.
* **RAM:** 16GB or higher for concurrent transactions and machine learning.
* **Storage:** 256 Solid State Drives (SSD) for swift data access.

**5.2 SOFTWARE REQUIREMENTS**

* **Web Development**
  + Web server software: Apache, Nginx.
  + Front-end Client: HTML, CSS, JavaScript.
  + Front-end Server: Python
  + Web Framework: Flask
* **Database Management System**
  + MySQL
* **Blockchain Platform:**
  + JSON
* **Machine Learning Framework:**
  + TensorFlow, PyTorch, or scikit-learn.
* **Programming Languages:**
  + Python for machine learning and backend.
  + Solidity for smart contract development.

**5.3 SOFTWARE DESCRIPTION**

**5.3.1 FRONT END: DEVELOPMENT**

**PYTHON 3.7.4**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

****

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard library which can be used for the following:

* Machine Learning
* GUI Applications (like Kivy, Tkinter, PyQt etc.)
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like OpenCV, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia
* Scientific computing
* Text processing and many more.

**Tensor Flow**

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications.



TensorFlow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages. Developers have the option to deploy models on a number of platforms such as on servers, in the cloud, on mobile and edge devices, in browsers, and on many other JavaScript platforms. This enables developers to go from model building and training to deployment much more easily.

**Pandas**

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.



Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda’s library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

**NumPy**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.



NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

**Matplotlib**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.



Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, python, Qt, or GTK.

**Pillow**

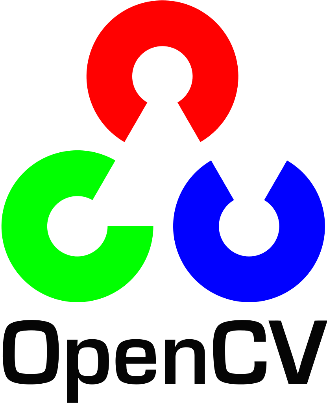
Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors.



Python pillow library is used to image class within it to show the image. The image modules that belong to the pillow package have a few inbuilt functions such as load images or create new images, etc.

**OpenCV**

OpenCV is an open-source library for the computer vision. It provides the facility to the machine to recognize the faces or objects.



In OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and video.

**5.2.2 MYSQL**

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.



MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

**5.2.3. WAMPSERVER**

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.



WAMPServer is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn’t require a payment or subscription.

**CHAPTER 6**

**RESULTS AND DISCUSSION**

The system’s performance during a cloud storage server disaster, by the HAEdge App Layer and VirtualBox, reflects a strategic approach to ensure swift data recovery and service continuity. Early detection of anomalies triggers the activation of the HAEdge App Layer, showcasing the system’s readiness for disaster recovery. The reliance on VirtualBox for archive retrieval underscores the importance of maintaining essential system images and configurations, ensuring critical data accessibility during the recovery phase. The initiation of data restoration from backups, facilitated by HAEdge, emphasizes the system’s commitment to preserving data integrity and leveraging redundancy to safeguard against potential data loss. The parallel activation of redundant systems by HAEdge ensures continuous data availability, minimizing downtime and ensuring uninterrupted service operation. Granting temporary data access to users through alternative channels, facilitated by HAEdge and VirtualBox collaboration, highlights a user-centric approach, maintaining productivity during the recovery process. Real-time communication with users transparently provides updates on the recovery process, managing user expectations and building trust in the system’s capabilities. Continuous monitoring, evaluation, and swift issue resolution demonstrate the system’s adaptability, ensuring that any deviations from the recovery plan are promptly identified and addressed. Data verification and integrity checks post-restoration emphasize the system’s commitment to maintaining the accuracy and reliability of restored data. The gradual resumption of normal operations ensures a controlled transition, minimizing potential disruptions and ensuring a smooth return to full operational status. Post-disaster analysis and documentation of lessons learned contribute to continuous improvement, informing future enhancements to the system’s resilience against unforeseen challenges. Clear communication signals the successful resolution to users, managing expectations, and confirming the restoration of normal operations. The iterative update of the disaster recovery plan based on gained insights reflects a proactive and adaptive approach, ensuring the system remains well-prepared for potential future disasters. In essence, the system’s performance during a disaster exemplifies a well-coordinated, user-centric, and continuously improving strategy for data recovery and service restoration.

**CHAPTER 7**

**CONCLUSION**

**7.1 Conclusion**

In conclusion, the integration of the HAEdge App Layer and VirtualBox in the system's response to a cloud storage server disaster presents a robust and proactive approach to safeguarding data and ensuring service continuity. The systematic flow of actions, from early detection to post-recovery analysis, underscores the system's resilience and adaptability in the face of unexpected challenges. The early detection mechanism, coupled with the swift activation of the HAEdge App Layer, demonstrates the system's preparedness for disaster recovery. Leveraging VirtualBox for archive retrieval ensures the availability of critical data, emphasizing the importance of maintaining essential system images and configurations. The initiation of data restoration from backups, guided by HAEdge, underscores the system's commitment to data integrity and redundancy. The parallel activation of redundant systems and the facilitation of temporary data access prioritize user-centricity, maintaining productivity during the recovery phase. Real-time communication with users ensures transparency and manages expectations, fostering trust in the system's capabilities. This system utilizes the advantages of Temporary-Cloud storage to ensure users' data protection and reliability and, at the same time, overcomes the problems of multi-Cloud using the Fog Computing paradigm. System users can easily and securely backup, restore, and modify their data without caring about the sophisticated operations to protect and secure the data on Temporary-Cloud storage. Extensive numerical results demonstrate the efficiency of the proposed scheme on improving survivability of data and services in cloud data centers. With a set of given resources and early warning time constraints, this project can guide data center operators to achieve a tradeoff between data backup and service migration.

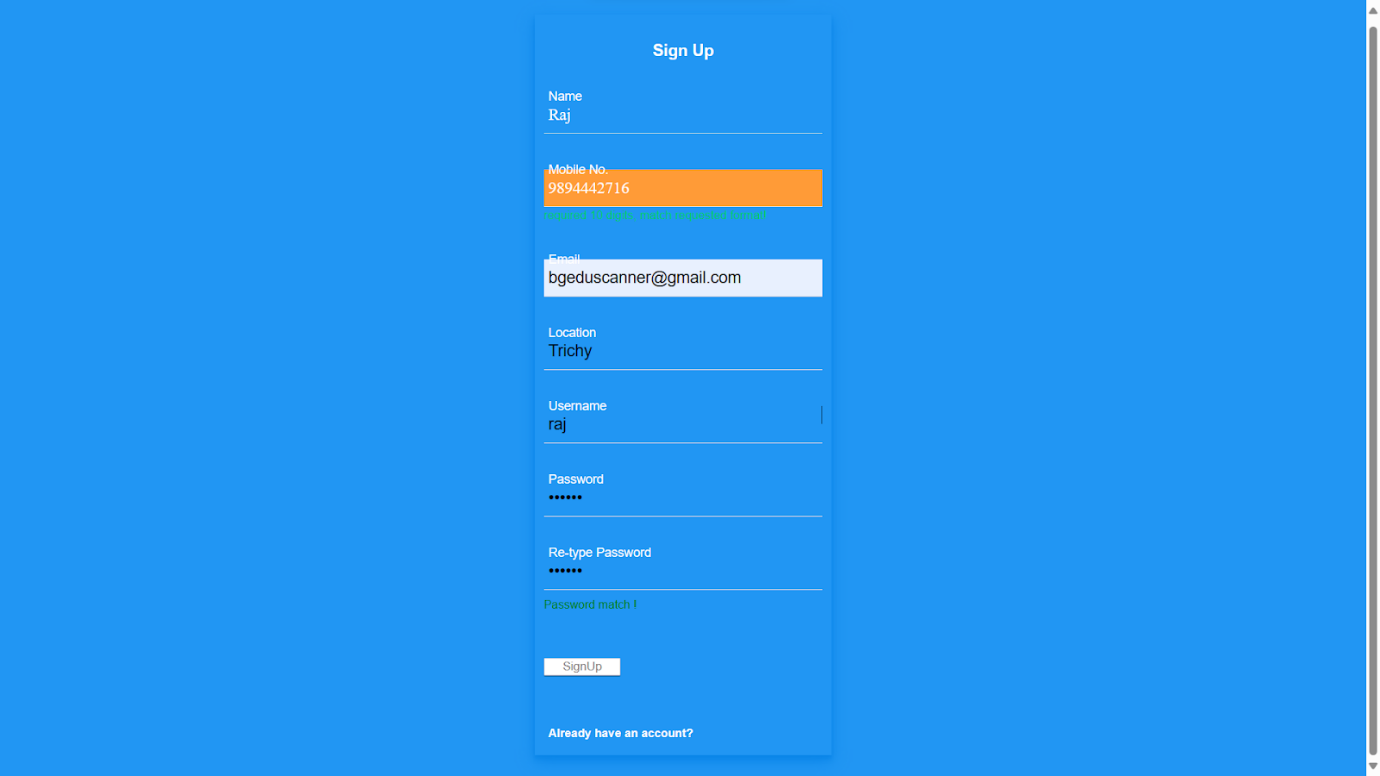
**7.2 Future Enhancement**

Future enhancements for the system, involving the HAEdge App Layer and VirtualBox, aim to bolster disaster recovery capabilities, elevate user experiences, and adapt to the evolving technological landscape. One prospective avenue for improvement involves the exploration of advanced machine learning integration. By incorporating sophisticated machine learning algorithms for predictive analysis and anomaly detection, the system could heighten its ability to anticipate potential issues, thereby enabling even faster response times during disasters. Another promising enhancement involves the consideration of blockchain technology for data integrity verification. Implementing blockchain can establish an immutable and transparent record of data changes, reinforcing the integrity of critical system images and configurations. This approach contributes to a robust and tamper-proof data verification mechanism. Furthermore, an essential aspect of future enhancement involves the implementation of a Zero Trust security architecture. This strategic approach ensures that trust is never assumed, and continuous verification of the security status of users, devices, and systems takes place. By adopting a Zero Trust model, the system can fortify its overall security posture, minimizing vulnerabilities and mitigating potential risks.

**CHAPTER 8**

**APPENDICES**

**APPENDIX 1 : SCREENSHOTS**



**Fig 8.1 Sign Up Page**

A screenshot of a blue box with white text

Description automatically generated

**Fig 8.2 Server Provider Sign in Page**

A screenshot of a computer

Description automatically generated

**Fig 8.3 Server Provider Dashboard**

A screenshot of a blue screen

Description automatically generated

**Fig 8.4 Login Page**

A screenshot of a computer

Description automatically generated

**Fig 8.5 Server Checking Page**

A screenshot of a computer

Description automatically generated

**Fig 8.6 ID Creating Page**

A screenshot of a computer

Description automatically generated

**Fig 8.7 Status Update Page**

A screen shot of a login screen

Description automatically generated

**Fig 8.8 User Sign in Page**

A screenshot of a computer

Description automatically generated

**Fig 8.9 User Received Files Page**

A screenshot of a computer

Description automatically generated

**Fig 8.9 Admin Login Page**

A screenshot of a computer

Description automatically generated

**Fig 8.10 Usage Checking**

**APPENDIX 2 : SOURCE CODE**

**Packages**

import os

from cryptography.fernet import Fernet

import numpy as np

import pandas as pd

from tkinter import filedialog

from tkinter.filedialog import asksaveasfile

from tkinter.filedialog import askopenfilename

from tkinter import messagebox

import cv2

import os

import time

import threading

import random

import tkinter as tk

from tkinter import ttk

import PIL.Image, PIL.ImageTk

from tkinter import \*

**Database Connection**

import mysql.connector

mydb = mysql.connector.connect(

  host="localhost",

  user="root",

  password="",

  charset="utf8",

  use\_pure=True,

  database="virtualbox"

)

**Open PGP Zip Compression**

def OpenPGP(rkey\_fpath):

    rsa\_pub, \_ = pgpy.PGPKey.from\_file(rkey\_fpath)

    rkey = rsa\_pub.subkeys.values()[0]

    filedoc = pgpy.PGPFile.new('my msg')

    encrypted\_file = rkey.encrypt(filedoc)

    print encrypted\_file.\_\_bytes\_\_()

def file\_compress(inp\_file\_names, out\_zip\_file):

    # Select the compression mode ZIP\_DEFLATED for compression

    # or zipfile.ZIP\_STORED to just store the file

    compression = zipfile.ZIP\_DEFLATED

    #print(f" \*\*\* Input File name passed for zipping - {inp\_file\_names}")

    # create the zip file first parameter path/name, second mode

    #print(f' \*\*\* out\_zip\_file is - {out\_zip\_file}')

    zf = zipfile.ZipFile(out\_zip\_file, mode="w")

    try:

        for file\_to\_write in inp\_file\_names:

            # Add file to the zip file

            # first parameter file to zip, second filename in zip

            print(f' \*\*\* Processing file {file\_to\_write}')

            zf.write(file\_to\_write, file\_to\_write, compress\_type=compression)

    except FileNotFoundError as e:

        print(f' \*\*\* Exception occurred during zip process - {e}')

    finally:

        # Don't forget to close the file!

        zf.close()

            df=[]

            fn=filename.split('.')

            filezip=fn[0]+".zip"

            fn1="static/upload/"+filename

            df.append(fn1)

            zip\_file="static/virtualbox/"+filezip

            file\_compress(df, zip\_file)

            filesize2=os.path.getsize(zip\_file)

**Open PGP  Zip Decompression**

def OpenPGP\_decompress():

    key, \_ = pgpy.PGPKey.from\_file('/path/private\_key\_file')

    with key.unlock('passphrase') as ukey:

        file\_name = 'encrypted\_file.pgp'

        # decrypt it

        enc\_content=pgpy.PGPMessage.from\_file(file\_name)

        clr\_content = key.decrypt(enc\_content)

fname = request.args.get('fname')

        fn=fname.split('.')

        filezip=fn[0]+".zip"

        #print(filezip)

        shutil.unpack\_archive("static/virtualbox/"+filezip, "static")

**Data Owner Registration**

<?php

session\_start();

include("dbconnect.php");

extract($\_REQUEST);

$rdate=date("d-m-Y");

if(isset($btn))

{

$mq=mysqli\_query($connect,"select max(id) from cu\_owner");

$mr=mysqli\_fetch\_array($mq);

$id=$mr['max(id)']+1;

$ins=mysqli\_query($connect,"insert into cu\_owner(id, name, mobile, email, location, uname, pass,rdate,status) values($id,'$name','$mobile','$email','$location','$uname','$pass','$rdate','0')");

if($ins)

{

?>

<script language="javascript">

window.location.href="reg2.php?id=<?php echo $id; ?>";

</script>

<?php

}

}

?>

**Upload File**

<?php

include("encrypt.php");

extract($\_REQUEST);

$uname=$\_SESSION['uname'];

$q1 = mysqli\_query($connect,"select \* from cu\_owner where uname='$uname'");

$r1=mysqli\_fetch\_array($q1);

$name=$r1['name'];

$loc=$r1['location'];

$rdate=date("d-m-Y");

$qq = mysqli\_query($connect,"select \* from cu\_admin where bcode='".$r1['bcode']."'");

$rr=mysqli\_fetch\_array($qq);

if(isset($btn))

{

$mqry = mysqli\_query($connect,"select max(id) as maxid from cu\_files");

$mrow=mysqli\_fetch\_array($mqry);

$id = $mrow['maxid']+1;

$pbkey=$uname;

$fileName="F".$id.$\_FILES['file']['name'];

$fsize=$\_FILES['file']['size'];

$ftype=$\_FILES['file']['type'];

$qq=mysqli\_query($connect,"insert into cu\_files(id,uname,filename,detail,rdate,filesize1,filesize2) values(".$id.",'".$uname."','".$fileName."','$detail','$rdate','$fsize','0')");

move\_uploaded\_file($\_FILES['file']['tmp\_name'],"upload/".$fileName);

$fname=$fileName;

$originalfile="upload/$fname";

$crypt="upload/$fname";

Cryptfile($originalfile,$crypt,$pbkey);

?>

**Share File**

<?php

include("encrypt.php");

extract($\_REQUEST);

$uname=$\_SESSION['uname'];

$q1 = mysqli\_query($connect,"select \* from cu\_owner where uname='$uname'");

$r1=mysqli\_fetch\_array($q1);

$name=$r1['name'];

$loc=$r1['location'];

$rdate=date("d-m-Y");

$q11 = mysqli\_query($connect,"select \* from cu\_files where id='$fid'");

$r11=mysqli\_fetch\_array($q11);

if(isset($btn))

{

for($i=0;$i<count($uid);$i++)

{

$q5=mysqli\_query($connect,"select \* from cu\_share where uname='$uid' && fid=$fid");

$n5=mysqli\_num\_rows($q5);

if($n5==0)

{

$mqry = mysqli\_query($connect,"select max(id) as maxid from cu\_share");

$mrow=mysqli\_fetch\_array($mqry);

$id = $mrow['maxid']+1;

$qq=mysqli\_query($connect,"insert into cu\_share(id, fid, uname, rdate) values(".$id.",$fid,'$uid[$i]','$rdate')");

}

}

?>

**REFERENCES**

1. Shivang Modi, Yash Dakwala and Vishwa Panchal, "Cloud Backup & Recovery Techniques of Cloud Computing and a Comparison between AWS and Azure Cloud", International Research Journal of Engineering and Technology (IRJET), vol. 07, no. 07, July 2020.

2. Abedallah Abualkishik, Ali Alwan and Yonis Gulzar, "Disaster Recovery in Cloud Computing Systems: An Overview", International Journal of Advanced Computer Science and Applications, vol. 11, pp. 702, 2020.

3. A.A.Tamimi, R. Dawood and L. Sadaqa, "Disaster Recovery Techniques in Cloud Computing", *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 845-850, 2019.

4. S. Anuprabha and M. Nivaashini, "Protection of Cloud Services from Disaster Using Recovery Mechanism with Openstack", 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1382-1387, 2018.

5. J. Mendonça, R. Lima, E. Queiroz, E. Andrade and D. S. Kim, "Evaluation of a Backup-as-a-Service Environment for Disaster Recovery", 2019 IEEE Symposium on Computers and Communications (ISCC), pp. 1-6, 2019.

6. S. Hamadah and D. Aqel, "A proposed virtual private cloud-based disaster recovery strategy", 2019 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2019 - Proc., pp. 469-73, 2019.

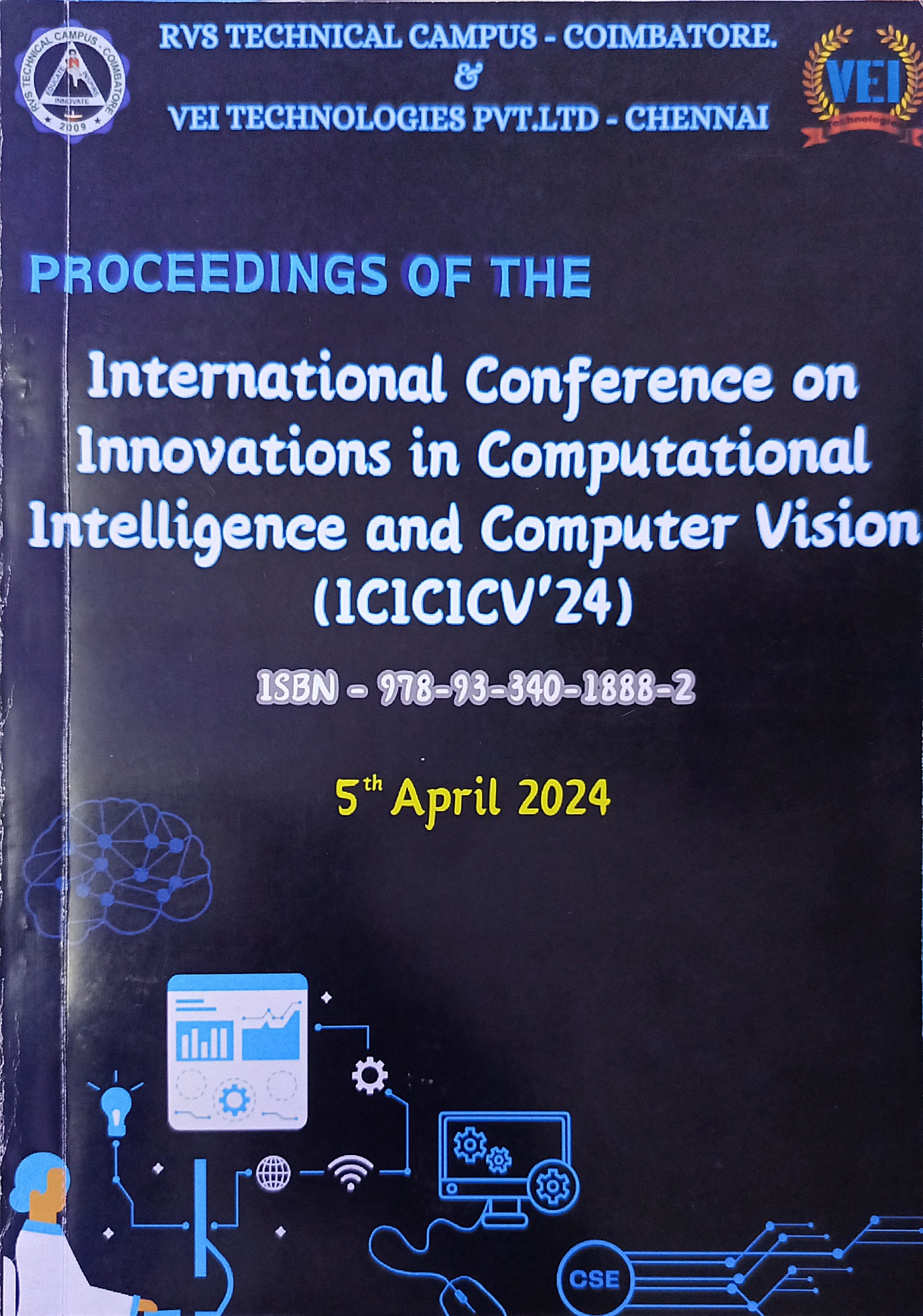
7. Bhalerao and A. Pawar, "Utilizing Cloud Storage for Big Data Backups", pp. 933-938, 2018.

8. M. S. Fernando, "IT disaster recovery system to ensure the business continuity of an organization", 2017 Natl. Inf. Technol. Conf. NITC 2017, vol. 2017-Septe, pp. 46-8, 2018

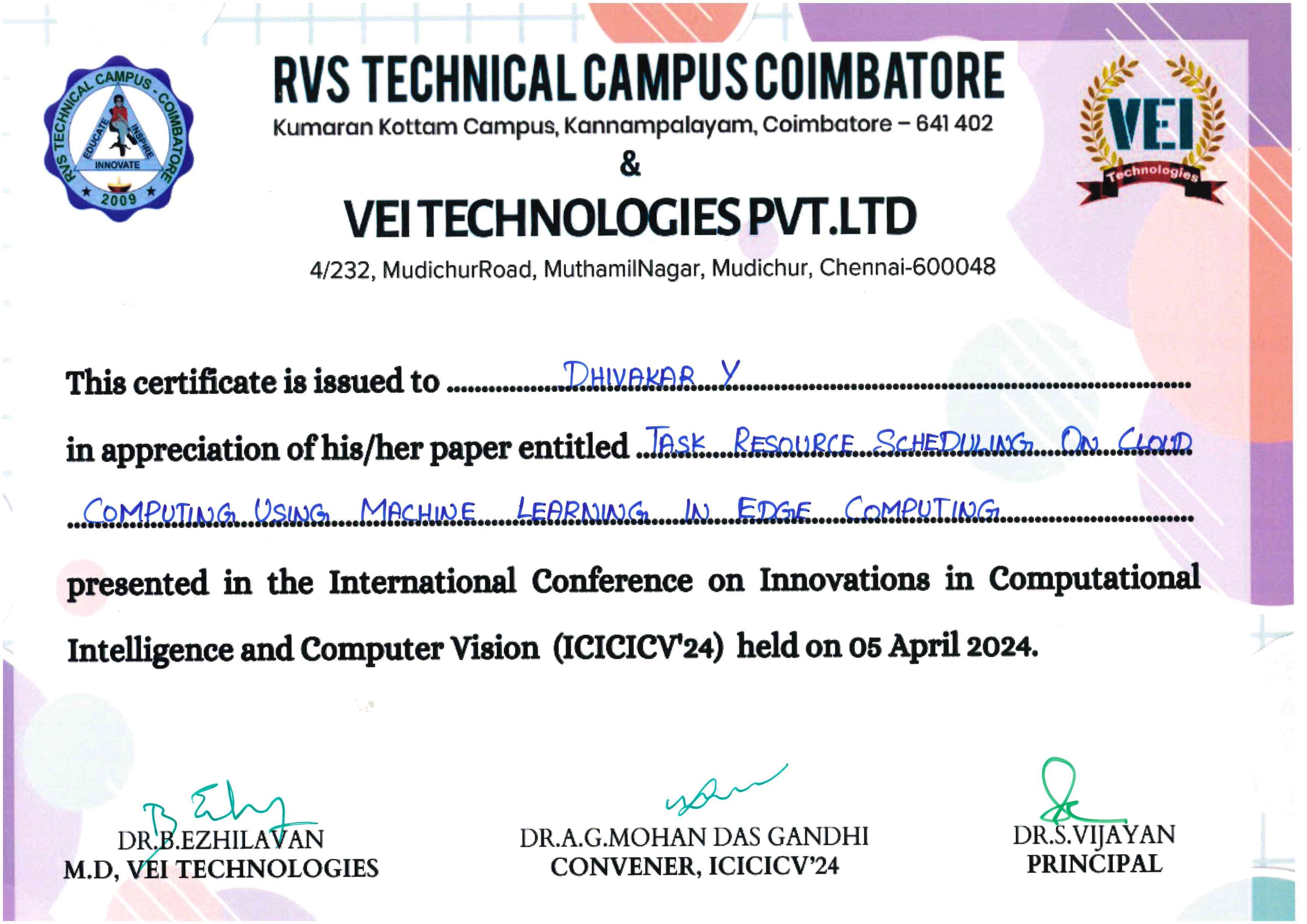
9. Mohammad Alshammari and Ali Alwan, "A Conceptual Framework for Disaster Recovery and Business Continuity of Database Services in Multi- Cloud", 2017.

10. Raigonda rani Megha and Tahseen Fatima, "A Cloud Based Automatic Recovery and Backup System with Video compression", International journal of engineering and computer science., vol. 5, pp. 17819-17822, 2016.

**CONFERENCE PUBLICATION**







A certificate of completion

Description automatically generated

