National College of Ireland

Developing Interoperable Blockchain Protocols for Secure Data Migration In Multi Cloud Environments

MSc Research Project

Saeed Adetugboboh
Student ID: 23212365

School of Computing
National College of Ireland

Supervisor: REJWANUL HAQUE

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. SAEED ADETUGBOBOH

Student ID: ……X23212365……..……

Programme: CLOUD COMPUTING **Year:** ……1……..

Module: ……RESEARCH PROJECT……

Supervisor: ……REJWANUL HAQUE……..……

Submission Due Date: ……..……

Project Title: ……Developing Interoperable Blockchain Protocols for Secure Data Migration in Multi-Cloud Environments

Word Count: ……9036…… **Page Count**…21…

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□

Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Developing Interoperable Blockchain Protocols for Secure Data Migration in Multi-Cloud Environments

Saeed Adetugboboh

23212365

Abstract

In a world increasingly reliant on multi cloud ecosystems, the easy, secure migration of sensitive data cross cloud service providers has become an important yet underexplored issue. Despite the promise of flexibility and cost efficiency, multi cloud strategies are held back by vendor-specific encryption standards, fragmented data management frameworks and the absence of interoperability protocols. This research gives a way to a blockchain driven way to tackle these pressing issues, introducing a decentralized protocol designed to secure the data migration while maintaining the compliance across the regulatory standards. While effective, the current implementation relies on static credentials, limiting its usability for multi-user environments. Future work focuses on integrating dynamic identity and credential management, enabling enterprise-level adoption, and enhancing the framework's ability to scale for larger datasets and complex workflows The research does not stop at technical validation, it lays the groundwork for wider practical applications. The findings establish not just a solution to the current migration bottlenecks but a vision for a decentralized, secure, and interoperable cloud future, where trust and efficiency converge easily

Keywords: Multi-cloud ecosystems, Data migration, Blockchain, Interoperability protocols, Decentralized protocol, Encryption standards, Regulatory compliance.

1 Introduction

The quick rise of cloud computing has changed and brought in new ways of how organizations manage, store and process data. Leading cloud service providers (CSP) such as Amazon Web Services (AWS) , Microsoft Azure and Google Cloud Platform (GCP) offer extensive scalability , flexibility and cost efficiency which acts as a cornerstone of the digital transformation , These services allow businesses to dynamically allocate different resources and streamline their processes easily. According to (Katie Costello, 2021) the global cloud computing market was projected to exceed $482 billion by 2023, driven by widespread adoption across industries. However, with the quick adoption of multi cloud strategies due to cost or other reasons, challenges have also increased especially in aspect of data migration and interoperability and security protocols. Data migration itself also has unique challenges, as it involves transferring encrypted or sensitive data between cloud providers, ensuring both security and data integrity. Research from (Mell & Grance, 2011) in NIST's "Cloud Computing Guidelines" talks more on maintaining encryption standards and compliance during migration as it is important to ensuring trustworthiness. However, migrating data which are encrypted

using vendor-specific algorithms shows an important technical and performance setback, further facilitating the need for secure and efficient protocols. The fragmentation caused by the proprietary encryption standards among the cloud providers further complicates the landscape. A 2021 report by Verizon documented an increase in data breaches which focuses on the risk associated to inconsistent protocols. (Verizon, 2021) These issues happen to be important in industries like the healthcare, where compliance with Health Insurance Portability and Accountability Act (HIPAA) ensures patient data confidentiality, and finance, where GDPR and Basel III standards necessitate strict data governance where the integrity and confidentiality of the data are important. (Buyya, et al., 2013) The architectures of multi cloud have become the norm rather than the exception for enterprises looking to optimize their costs and their operational performance. A 2024 report by Flexera found that 89% of enterprises adopt multi-cloud strategies, with an average of five different cloud providers in use. This rapid rise of cloud providers underscores the need for interoperable solutions capable of bridging technological and operational gaps. However, current practices often involve ad hoc solutions that lack the scalability and reliability required for enterprise-grade deployments. (Tošić, et al., 2023) The absence of standardized protocols among cloud service providers makes the challenges of multi-cloud management worse. Each Cloud provider implements proprietary encryption and data management frameworks, resulting in siloed data that is difficult to integrate or migrate. This fragmentation not only hampers operational efficiency but also raises significant security concerns, as the inconsistent encryption mechanisms can leave data vulnerable to breaches. Addressing these issues requires a unified approach that incorporates robust encryption, seamless interoperability, and real-time verification of data integrity during migrations. Which brings up new technologies like blockchain and homomorphic encryption, which present new opportunities to address these challenges. Blockchain, with its immutable and decentralized architecture, can facilitate real-time verification of data integrity during migration, while homomorphic encryption offers a means to process encrypted data without decryption. (Zhang, et al., 2010) By taking advantage of such innovations, organizations can improve their security, ensure compliance, and improve their trust in multi-cloud environments.

1.1 Research question

This research addresses the following question:

How can interoperable blockchain protocols facilitate secure and efficient data migration across multi-cloud environments while maintaining encryption integrity and compliance?

The quick adoption of the hybrid cloud or multi cloud strategies reflects the huge shifts toward more versatile IT solutions. In the beginning of 2024, 89% of enterprises picked up these strategies, managing an average of five cloud platforms. (Flexera, 2024) Moreover, like discussed in section 1, challenges will rise due to these adaptations. Gartner predicts that by 2025, over 75% of large organizations will face issues that will be related to cloud interoperability, with inadequate data migration strategies costing businesses an average of $5 million annually in downtime and recovery costs. (Gartner, 2023) These statistics really show the urgency of developing solutions that address the inefficiencies of current systems. The

existing data migration processes are complicated by the nature of encryption frameworks used by different cloud providers, for example AWS uses KMS (Key Management Service), which offers encryption that are tied to specific regions, while GCP uses its Cloud Key Management service, creating potential conflicts when migrating encrypted data across the providers. These limitations are governed by regulatory requirements such as GDPR, which imposes heavy penalties for non-compliance, amounting to up to €20 million or 4% of annual global turnover. (Parliament, 2016). Not only are there compliance issues, security also adds another dimension to the problem. Verizon's investigation report showed that over 45% of cloud-based breaches occur due to misconfigurations and lack of encryption consistency during data transfers. In the healthcare sector, breaches in data integrity during migration could compromise patient records, violating HIPAA guidelines and risking fines which could be up to $50,000 per violation. Similarly, in finance, data migration errors can disrupt real-time trading operations, leading to financial losses and reputational damage and this is where blockchain comes in to offer a new creative solution for these challenges because of its decentralized nature making a secure framework. Cryptographic hashing ensures that even the smallest data differences are flagged, while smart contracts can automate the compliance checks during migrations. A 2023 study by MarketsandMarkets projected that the blockchain market in multi cloud environments will grow at a CAGR of 67.3 from 2021 to 2028, which we are seeing to be true these days as it reflects its importance in addressing interoperability challenges. (MarketsandMarkets, 2021) The research aims to look at the effectiveness of blockchain-based protocols in improving the security of multi cloud migration.

1.2 Research Objectives

- Investigate the impact of vendor-specific encryption algorithms on interoperability.
- Examine the implications of inconsistent compliance frameworks on data integrity.
- Design interoperable protocols to ensure secure and verifiable data migration.
- Implement real-time data integrity checks using cryptographic hashing and blockchain verification.
- Measure the efficiency and security of the proposed solution compared to traditional migration approaches.

2 Related Work

These days the growing adoption of cloud computing has led to more complex and difficult data management challenges, particularly when it comes to moving data between multiple cloud providers. The traditional solutions had not evolved enough to ensure secure, traceable and interoperable data migration. In response, different studies have explored how blockchain, as a decentralized and immutable ledger, which can improve the data integrity, security and traceability in hybrid cloud environments. This section shows new advancements in this field,

focusing on frameworks that implement the blockchain for a more secure data migration and integrity verification in hybrid cloud and edge computing scenarios,

2.1 Blockchain for data integrity verification

Blockchain technology is evolving as a reliable tool for data integrity verification, especially in hybrid or multi-cloud scenes where the security risks arise from the decentralized nature of data storage and transfers. In cloud environments that works with multiple providers, maintaining a good level of data integrity and security becomes an issue, as each provider may implement various security measures of their own, gaps tend to be left in protection of the data during the data migration. (Witanto, et al., 2023) showed a decentralized integrity verification framework that addresses this gaps and challenges. Using blockchain as an immutable platform for data validation checks across the providers. Their framework takes advantage of blockchain decentralized architecture to show distribute integrity checks across multiple verifiers, thereby reducing the risk of any single point of failure while improvng interoperability and way of tracing data on the chain unlike the traditional approaches which are limited in multi cloud contexts since data migration between cloud adda numerous points where unauthorised access or data tampering could occur. By making the verification process decentralized , (Witanto, et al., 2023) states blockchain gave a good structure for interoperability , making sure that once data is verified on the blockchain, the verification was accessible accrosss all participating clouds without additional requirements of integration. A good feature which was proposed in the framework is its multi verifier design. Rather than relying on a single entity to verify the data , verification was enabled by different entities within the blockchain network, which improved the frameworks resillence against both internal and external threats. The multi-verifier systems by distributing validation authority , reducing exposure to cyber attacks especially those that target a specific cloud providers infrastructure, the transparency of the blockchain shows a method for real-time validation that keeps verifiable logs of data migration.

2.2 Decentralized application migration on edge

Surprisingly the integration of decentralized protocols for application migration on edge devices has transformed data management , especially in environments requiring real-time responsiveness, security and computational operative workload. (Tošić, et al., 2023) introduces a blockchain protocol for real-time application migraton suited for edge devices, focusing on handling the migration in resource constrained environments like Internet of Things (ioT) and mobile edge computing. Their study , talks more on blockchain potential in decentralized verification processs during migration, which improves data security and reduces the latency which is important during edge operations which require instant data processing.

The protocol developed by (Tošić, et al., 2023)is designed to address the core challenges such as limited computational power, quick connectivity and the need for fast, autonomous decision made on edge nodes. The traditional centralized models and dependencies on a single point of control show risks of latency and data integrity issues, particularly for edge applications like connected cars, remote monitoring and IoT devices. By making the data verification

decentralized using blockchain-based protocol, the framework made sure the validation and integrity is protected with the unchangeable audit trail. The use of the lightweight consensus mechanisms was a brilliant idea which was suited for resource constrained devices, achieving a middle balance between the security and operations that traditional blockchain models may struggle with edge scenarios. However, (Kanagachalam, et al., 2022) showed earlier works in environments with low bandwidth and fluctuating connectivity like automotive systems and mobile apps. Their protocol BloSM (Blockchain-Based Service Migration), showed secure and real-time service migration between cloud and edge in tight environments which addressed challenges like processing issues and data integrity in disconnected or low connectivity conditions. The protocol they worked on made the management of the events decentralized across connected car networks, making sure the applications can move between the nodes easily as vehicles travel across different network zones. This model of autonomy supported blockchain based verification reducing the risk of data loss or tampering during network transitions which is a big issue in connected automotive system where continuous operation and fast data verification are important for safety and functionality. These studies made by the researchers really showed the importance of decentralized verification for edge applications showing blockchains promise in supporting multi-cloud, edge-integrated environments especially when it involves high demands for real-time data processing. Innovations made like this pave the way for a stronger and efficient systems in areas like the healthcare, where continuous data migration and verification are critical in-patient monitoring and disaster management where reliable and fast data flow could be life saving. (Khan, et al., 2023) By bridging the gap between cloud and edge computing through blockchain protocols these researchers set the stage for deeper explorations into using blockchain protocols to solve solutions in hybrid cloud environments.

2.3 Decentralized data migration

Couple of researchers gave a wide solution aimed at securing data migration from centralised cloud environments to decentralized storage systems , (Khan, et al., 2023) implented the idea of blockchain ledger technology being in the central role of the framework buy enabling the traceablility of data changes. The inctive feature of (Khan, et al., 2023) approach is its integration of blockchain with decentralized storage such as IPFS (InterPlanetary File System) by moving the cloud storage to decentralized storage. The model makes syre the data remains accessible, verifiable and strong against unauthorised tampering. The IPFS integration joined with blockchain verification, allows for a high degree of interoperability between different storage systems and further reinforces data availability, even in the event of parytial network failures. Perhaps their work shows the role of blockchain as a universal protocol layer which resonates well in showing blockchain can unify different cloud and decentralized systems under a single secure migration framework. This has been a great foundational in this research as it creates a way for using the blockchain not just as a security measure but as an interoperable platform that can adapt to various cloud infrastructures.

3 Research Methodology

This section details the research process behind the designing and implementation of the interoperable blockchain protocol or secure data migration across hybrid cloud environments. The methodology includes an in depth discussion on the systems architecture and tool selection, security mechanisms that were integral to making sure the data confidentiality and deployment and testing processes. The approach was informed by prior research made and best security practices made to secure data handling and the unique demands of interoperability across the cloud platforms.

The systems architecture was designed to achieve easy data migration across three primary cloud providers, Amazon Web Services, Microsoft Azure and Google cloud- integrating blockchain for the metadata tracking. The cloud provider required an individualized approach to integrate the data storage, key management and access protocols while also preserving the interoperability between the providers. The blockchain component however was managed through smart contracts written in Solidity and Remix IDE was chosen for the developing and testing of the contracts due to the ability to be compatible with Ethereum which is known as the world's computer. The design was made to facilitate the migration and store the metadata of each file's integrity status improving the traceability and security. The architecture was selected to support the easy flow of data between cloud platforms, using the advantage of Application Programming Interface (API) endpoints for communication. AWS S3, Azure Blob Storage and Google cloud Storage were selected as the storage services due to their scalability, API support and industry standard in multi cloud project. However, for the smart contract, Ethereum was chosen for its decentralized and popular nature allowing a transparent, supported and immutable metadata storage to record and receive file integrity during the migration, a smart contract was created with three functionalities which are to perform the metadata storage, migration event recording and hash verification. This enabled storage of each file's integrity information. In choosing tools, specific design considerations were made. First, API integrations were essential, as each of the cloud provider offers unique API specifications. AWS S3 was accessed via the AWS software development kit, which was configured to handle encrypted uploads and retrievals with secure access management. Similarly the Azure blob storage and google cloud storage APIs enabled the secure object storage through the software development kits, while the key management services were employed to encrypt the data encryption keys across the individual clouds achieving the cross cloud compatibility and security.

3.1 Data Collection

The data collection methodology is focused on the framework's ability to manage and securely migrate various types of data across multicloud environments rather than on specific datasets. Given the project's goal of developing an interoperable blockchain protocol for secure data migration, the framework is designed to support a wide range of data types, including text files, images, PDF, csv files rather than a fixed dataset. Making the emphasis on making sure the

protocol can handle any data type and sizes securely and efficiently during migration processes and sample data files ranging in size from a few kilobytes to several megabytes were used to test the encryption , hashing and migration . This makes the ability to examine the protocols adaptability and efficiency when handling different data structures which gives the advantage of allowing the framework to be universally applicable across different use cases in multi cloud environments , where the users get to work often with heterogenous data.

3.2 Encryption and Hashing methods

The development of a secure data migration protocol for hybrid cloud environments, encryption and hashing were really important to maintaining the data confidentiality and integrity across the different stages of migration. The details of the cryptographic principles , key management strategies and security protocols uses the AES-256 encryption and SHA-256 hashing and Key management system integrations from the cloud providers. Each of the cryptographic method is well grounded in proven researcha nd practical applications in cloud security which has been referenced from seminal texts in the field. The security of the data migration in multi cloud environments depends on wide and strong encryption mechanisms. This project however adopted the Advanced Encryption Standard known as AES spefically, the AES-256 as it is known for providing high level security due to its 256-bit key length. As (Anderson, 2001) emphasiz3es about how AES is a critical algorithm in securing distributed systems. The AES-256 follows a symmetric encryption approach, where the same key is both used for the encryption and decryption process. The process involved several stages including key expansions , subBytes transformation , shiftRows and Mix Columns. The key expansion step however based on Rijndael's key schedule uses XOR operations and S-box transformation to generate sub keys for each encryption round which improves the algorithm resillence against brute force attacks (Daemen & Rijmen, 2002). subBytes however applies a non linear S-box transformation making sure that each byte of the plain text infulences the cipher text output. According to (Daemen & Rijmen, 2002) analysis, the shiftRows and mixColumns operations ensure the diffusion by spreading the bits from the original key across the transformed cipher text. These transformations were fundamental for resisting the differential and linear cryptanalysis , making the AES-256 the preferred method to go with for the cloud security. The encryption round itself concludes with the AddRoundKey step , where the subkey is XORed with the transformed state matrix. This operation performed in every round, interweaves the encryption key into the data at multiple stages , improving rhe data security. The encryption process which happens to be mathematically expressed as

$$C=E\ (K\ ,P)$$

Whereas the C represents the ciphertext, K the 256-bit encryption key and P , the plain text/ The reversible nature of AES-256 allows the data decryption with the same key, an important feature for securely retrieving the data after migration. which (Anderson, 2001) validates when AES-256 role comes into play in securing the cloud-based data especially when the confidentiality of the data is quite important in multi cloud infrastructure. Even if the

encryption protects the data, the data integrity still becomes questionable especially during migration as the owners of the data gets concerned about the tampering or loss of contents, which made cryptographic hashing a necessary component and method to follow. The Secure Hash Algorithm (SHA-256) which is also a widely adopted hash function was used in this project to generate a unique hash for each file. SHA-256 produces a 256 bit hash, which acts as a digital fingerprint of the data and also becomes important for establishing high data integrity in cloud environments (Winkler, 2011). The SHA-256 tends to apply bitwise operations and modular additions to achieve nonlinearity, which makes it resistant to collision attacks, which happens to be an important requirement for secure hashing. It compresses the data in 512 bit blocks, processing each of the blocks through multiple rounds involving the modular arithmetic and logical shifts. This design makes sure that even the smallest change to the data produces a drastically different hash, a phenomenon known as the avalanche effect. (Damgård, 2001). The deterministic property of SHA-256 represented as

$$H\,(M) = SHA\text{-}256\,(M)$$

Where H(M). represents the hash output for message M. SHA-256. This makes it ideal for verifying the data integrity in cloud environments as it ensures the hash remains consistent if the data remains unaltered (O'Hara & Ben Malisow, 2016). This method of maintaining integrity with the hash makes sure the file is preserved during the transition between the cloud environments. However the encryption and integrity also needs to have a key management in place to avoid unauthorised accesses using the Key management systems of each of the top cloud service providers, Amazon web Services, Google and Azure (Kavis, 2014) as it is adopted to use their cloud native services to manage cryptographic operations within their environments to reduce potential vulnerabilities and increase trust (Winkler, 2011). By following this method, the data encryption key gets encrypted at the origin cloud and making it possible to be decrypted only at the destination cloud, the Key management system protocol makes sure that the data cannot be accessed without the unauthorised keys. It also makes it a strong combination of security and encryption model for cloud data as the data encryption keys is encrypted before transmission making the data packet unintelligible even when intercepted. (Schneier, et al., 2011) achieving data security and integrity across the cloud migrations.

3.3 Blockchain hashing method

The blockchain technology which happens to be structured as a chain of blocks , where each block contains data , a time stamp and the cryptographic hash of the former block makes it resistant to tampering , as modyfying a block requires recalculating the hashes of all chained blocks across the network, a practically impossible task (Nakamoto, 2008). However, in the context of blockchain , it gets to be used to protect the linking of blocks in the chain rather than directly protecting the users data. each block's contents, which includes a reference to the previous block's hash, are hashed with a function like SHA-256, creating a unique identifier for each block. The relationship between blocks are represented as

$$H\,(B_i) = SHA256\,(\,data_i \,||\, H\,(B_i{-}1))$$

where $H(B_i)$ is the hash of the current block i, $data_i$ represents the transaction data within the block, and $H(B_{i-1})$ is the hash of the previous block. This chaining process shows that any attempt to change the data in one block will invalidate the hashes in all subsequent blocks, ensuring the integrity of the entire chain. (Dinh, et al., 2018). In essence, while the blockchain SHA-256 secures the structure of the ledger itself, SHA-256 hashing of files ensures the integrity of the data being migrated. Together, these applications of SHA-256 bring up a multi-layered security framework where the blockchain verifies the integrity of migration records, and file-level hashing verifies data integrity within those records, making the system highly resistant to fraud and data corruption.

4 Design Specification

The design specifications for this research project gives a strong and advanced multi cloud data migration architecture secured with encryption and blockchain-based verification, which supports the ability of interoperability across leading cloud providers. This solution also allows the data to be properly encrypted, transferred and verified across Amazon Web Services, Microsoft Azure and Google Cloud Platform which makes sure the data integrity and security is high and consistent. The proposed architecture takes advantage of three core cloud service providers which offer its own storage and key management system (KMS) which the architecture integrates to make sure it is secured. This architecture is centred to the use of the cloud providers specific KMS which makes sure that each cloud service providers are independently managing their encryption keys to prevent unauthorised data exposure. Where when the files get uploaded to the CSP's they are immediately encrypted with a Data Encryption Key (DEK) which gets to be generated by the cloud provider's KMS. The encrypted file and its metadata which includes the encrypted DEK and an initialization vector for the encryption are then uploaded to the cloud provider's storage. For migration, the files are securely downloaded, decrypted and then re-encrypted using the target cloud provider's KMS before it gets uploaded to the new destination. The blockchain however records each file's metadata like the file hash, cloud provider, timestamp to make sure the file's integrity is maintained to provide a secure, immutable log of file storage and migration history. This setup offers not only a secure but interoperable framework where the data migration between the cloud providers is achieved with high level of security, traceability and data integrity.

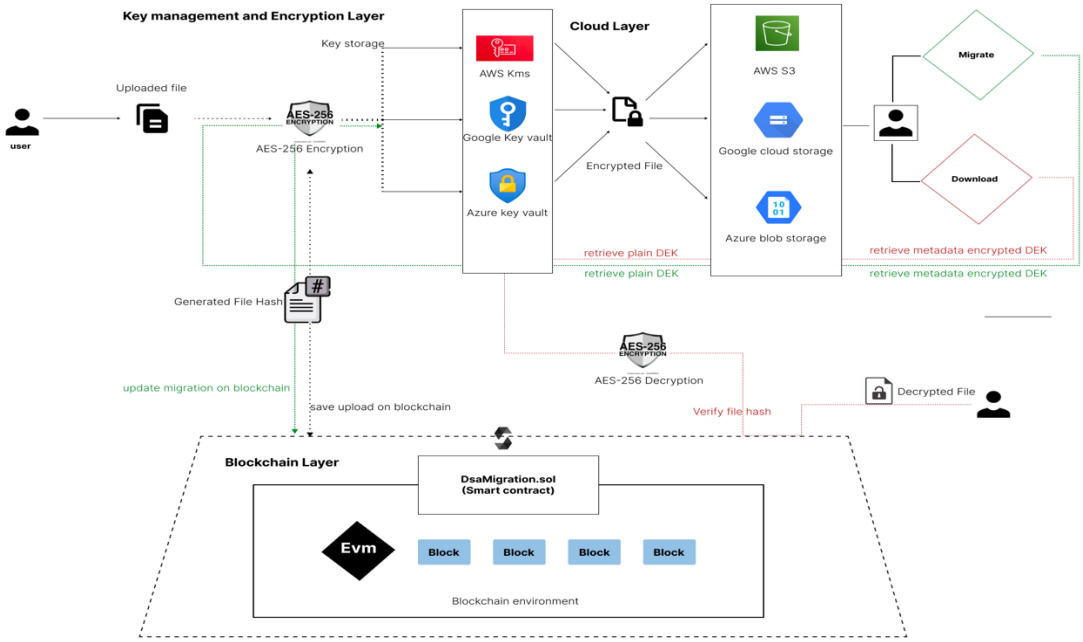

Fig 1. Architectural diagram

5 Implementation

The implementation of this project brings the design specification to life demonstrating the practical realization of secure data migration across multiple cloud service providers using blockchain technology. This section details each stage of the systems deployment configuration focusing on how the integration of the blockchain smart contracts was made to make the metadata logging, encryption and decryption process stronger and the JavaScript client interface used to manage these operations. Together, these elements form a secure and interoperable framework for the cloud migration.

5.1 Blockchain and Smart contract Deployment

The implementation begins with the setting up of the blockchains component to manage file metadata securely and transparently. Smart contracts got to be coded in solidity, Ethereum's native programming language and deployed on the ethereum blockhain, The two primary functions in the smart contract are `addFilemetadata` and `recordMigration`. The addFileMetadata functions records the hash and storage provider metadata of each uploaded file , while the recordMigration function logs the migration details, including the source and destination cloud providers and the migration timestamp. This follows the principles established by (Dinh, et al., 2018) who demonastrated the importance of verifiable data records in decentralized applications using blockchain technology like ethereum which is chosen for its wide abiliity to support smart contracts and widespread adoption around the world which increases the systems interoperability. The deployment of the smart contract is conducted by using a framework called remix which facilitated testing and deployment to the ethereum network . Each function in the smart contract was designed with gas efficiency in mund using data types that woul avoid difficult computations while transacting. Gas techniques however

ensures that each operation whether recording file metadata or logging a migration event, which executes at an affordable and predictable cost range. This gas optimization however is crictical as it ensures that the decentralized nature of the blockchain does not become an issue for system performance or scalability. However the interaction with the blockchain from the client side javascript application was made possible via the web3.js to establish a connection to Metamask, allowing the users to initiate blockchain transactions directly from their browsers. Metamask, which is a popular ethereum wallet allows the users to manage the accounts and authorise the blockchain transactions allowing a easy interface for the data migration process. Once the user uploads a file or initiates a migration , the application invokes the appropriate smart contravt function making a blockhcain record of the event , however each transaction gets to be signed by the user's metamask account ensuring that each action on the blockchain is authenticated and traceable.

5.2 Cloud Storage Bucket Configuration

To also enable the multi cloud data migration, the storage buckets were configured on Amazon simple storage service , Azure Blob storage and Google Cloud Storage. These platforms were chosen for their diffffernt distinct advantages in reliability, security and global accessibility , aligning with the project goals of cross providers operability and data security. Each of the storage buckets serves as a secure container for files in transit, with storage classes which are optimized for frequent access and retrieval. Amazon simple storage however was selected for its flexible , highly scalable nature, which accomodates fluctuating workloads without sacrificing its performances. Not only does it have an advantage in that , the multi region support further makes sure that the data gets to be distributed across different geographical locations , which is really an important factor for compliance and latency optimizations. Azure blob storage on the other hand was chosen for its native support for unstructured data and ability to be compatible with a wide array of data typwes ensuring higher support to the data management. Google cloud storagge however was incorporated for its powerful data processing integrations , such as big query which facilitates future data analytics possibilities. This selection is basically supported by the wide study of (Zhang, et al., 2010) who talked about the advantages in their analysis of cloud storage solutions for distributed systems bucket to be configured withh strict identity and Access Management policies to linmit the access to authorised users only, making the security improved at this stage of the migration as well. AWS IAM roles were set up to permit the access to these s3 buckets , Google IAM roles were also configured for the Google cloud storage and Azure Active Directory roles for Azure. These roles were applied to make sure that the applications's services could read and write data to the storage buckets aligning with the principle of least priviledge in security devops. Furthermore , the encryotion policies with the cloud providers were enforced to automatically encrypt the file at rest ensuring data confidentiality, making it impossible to download directly from the cloud provider.

5.3 Encryption and Key Management

Each cloud service provider use its own key management system just as advised (Barker, 2020) This key management system generates and store the Data Encryption Keys needed to encrypt the data. This DEK is used for the AES-256 encryption of each file , which makes sure that data is protected during the storage and transit. Upon the file upload , a unique DEK gets generated and encrypted by the KMS and gets to store the metadata and the data associated with it in the cloud storage. The initialization vector is also generated and stored with the files metadata to make sure the decryption of the file is still possible even if the file migrates to a different cloud provider. For the file retrieval and migration, the system gets to retrieve the encrypted DEK and Initial vector from the file meta data which the KMS decrypts , allowing the file content to be decrypted and rencrypted as necessary. Although these values are important to decrypting the file content and also concerned about it getting into the wrong hands of cyber threats, these values are not sufficient on their own. The DEK itself is encrypted by the key management system which only the KMS can decrypt ensuring that unauthorised users cannot decrypt the file by simply having access to the DEK and initialization vector. Even if an anauthorised party gains access to these encrypted keys, without the proper permissions granted through the KMS, they cannot initiate the decryption process. Through this approach the data remains encrypted at all stages, both at rest and in transition during the migration between cloud platforms. This setup follows strict data security standards for multi cloud environments by making sure that only verified and unauthorised parties can access or manipulate the data even if they have access to the files metadata and storage location

6 Results and Evaluation

The result and evaluation phase is quite important to validate the effectiveness, security and performance of the developed framework. This section however examines the proposed blockchain based data migration framework for hybrid environments, focusing on the core aspects, the security, integrity , performance and interoperability across the different cloud service providers. The main objective however is to assess how well the system meets the specified requirements and whether it successfully addresses the challenges identified in the initial problem statement and making sure key metrics such as the encryption integrity, data migration efficiency, blockchain transaction integrity and system interoperability are tested to make sure the protocol is reliable

.

6.1 Security and Data integrity Evaluation

The security is a central component of this research as discussed earlier in previous sections given that it involves secure data migration across the cloud environments. This part of the evaluation looks at the steps to upload migrate and verify file data integrity using cryptographic methods that show successful implementation, metadata consistency across the platforms and blockchain based verification.

6.1.1 File upload and Initial Hash computation

In the system, each file uploaded goes a SHA-256 hashing process immediately upon the upload as shown in figure 2.1 and as discussed about how the SHA acts as a fingerprint for the file (Damgård, 2001). This unique hash generated is important for ensuring as any tampering or alteration made to the files content will produce a different hash value. To further validate the accuracy of the hash computation the same file was uploaded to an independent SHA-256 hash generator available online. The output from this trusted third party tool confirmed that the system generated hash matched up with the hash produced showing the reliability and correctness of the hashing implementation in figure 2.2 This cross-verification however assures the users and stakeholders that the system's hash computation follows the standard cryptographic practices.

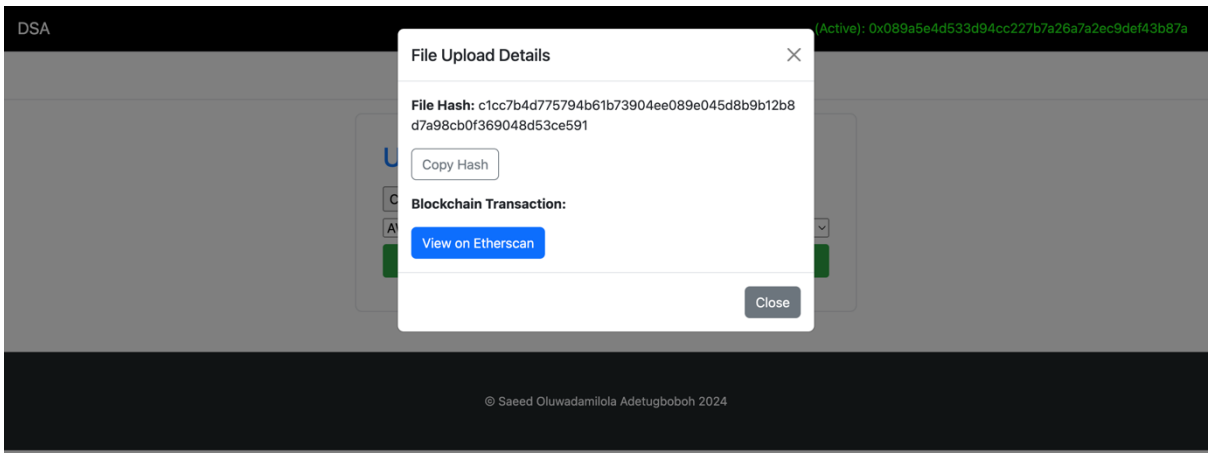

Fig 2.1 Confirmation of Successful File Upload with Computed Hash

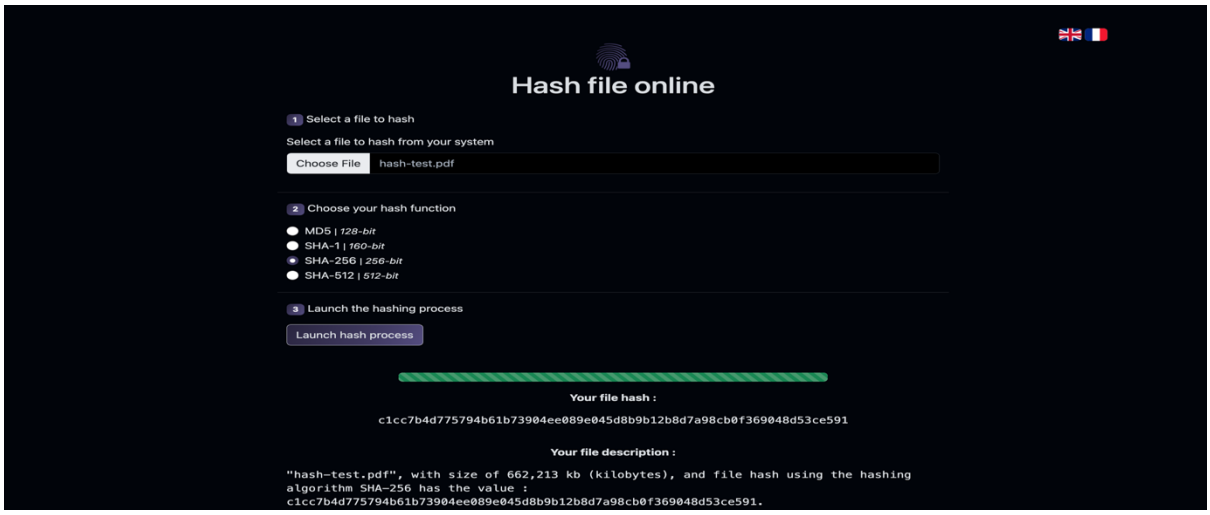

Fig 2.2 - Independent Verification of SHA-256 Hash with Online Hash Generator

6.1.2 Metadata Preservation on Amazon Web Services

After the successful upload, the important metadata, including the original SHA-256 hash, encrypted Data Encryption Key and Initialization vector is stored alongside the file in AWS as

shown in figure 3.1 in which according to (Anderson, 2001), securely storing the cryptographic keys and metadata with encrypted data gets to ensure that unauthorised parties cannot access the data without possessing the required key and decryption parameters needed. However, the original hash in the AWS metadata further supports the integrity verification which aligns with the best security practices (Winkler, 2011), as the hash value in the AWS can later be cross referenced to make sure the data doesn't get tampered with during the migration.

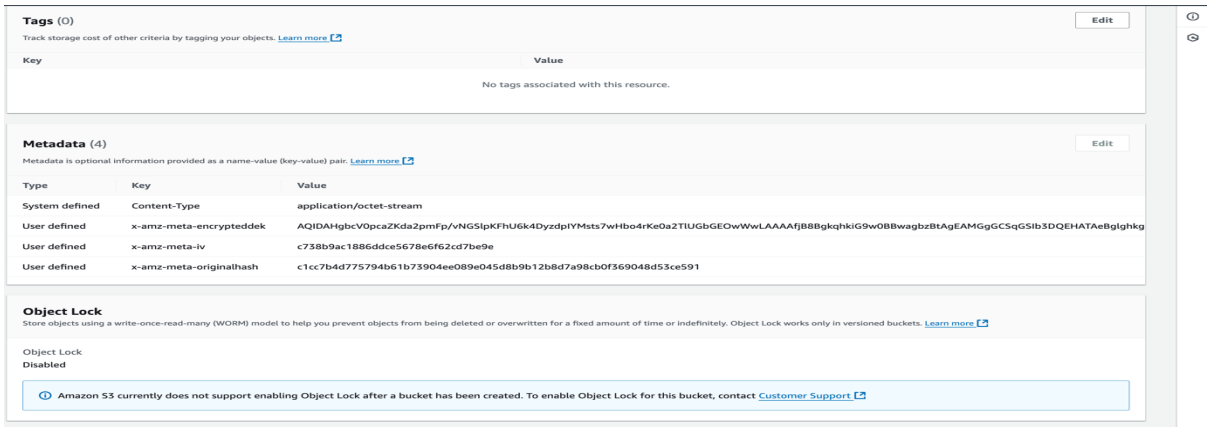

Fig 3 - AWS Metadata Showing Original Hash, Encrypted DEK, and IV

During the migration to Google Cloud Platform, the system maintains a metadata consistency in figure 4 by transferring the exact same metadata as seen in figure 3.1 from AWS. This metadata transfer is important as it preserves the cryptographic foundation of the file across the cloud environments (O'Hara & Ben Malisow, 2016). This however shows the interoperability between the two cloud service providers allowing the proper storage and retrieval of cryptographic data regardless of the cloud providers. This cross-platform metadata integrity upholds the security model described by (Kavis, 2014) where cloud providers are treated about interchangeable components in a secure architecture.

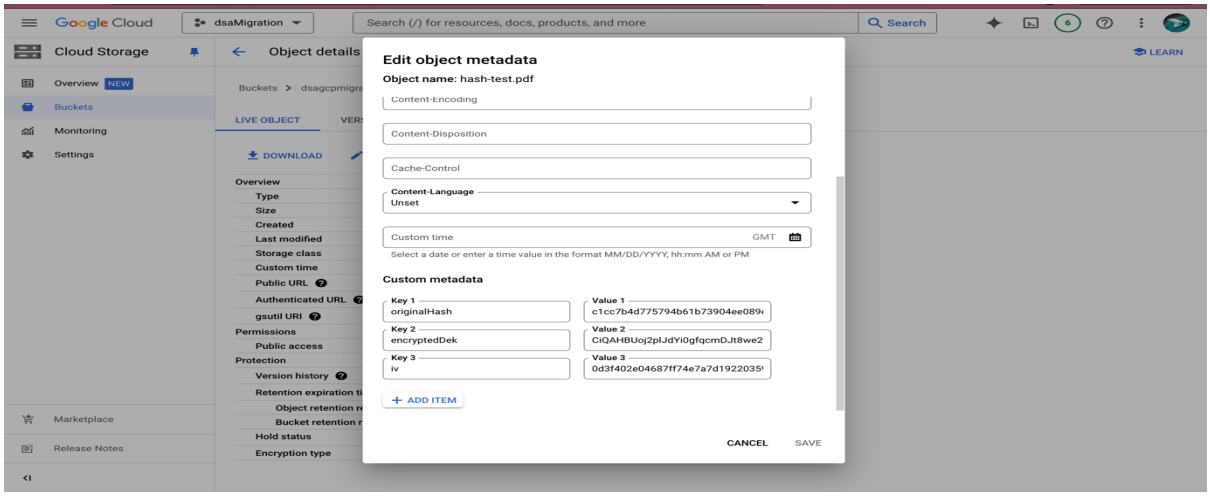

Fig 4 - GCP Metadata Displaying Transferred Original Hash, Encrypted DEK, and IV

6.1.3 Blockchain-Based Verification and Cost Analysis

Regardless of the SHA-256 integrity-based verification of the file hash, the whole transparency of the system is recorded in the Ethereum blockchain, This addition makes users to verify their file authenticity independently , using the ability of blockchains immutability to protect against data corruption and tampering (Nakamoto, 2008) by recording the metadata on a decentralized ledger the system then makes a permanent, verifiable record, increasing the user confidence in data integrity. The metadata that is added to the blockchain not only ensures the data migration event is immutable but also enables users to audit the metadata on the blockchain explorer as shown in figure 5.1. As the transaction gets confirmed, it is shown on Etherscan, which allows the verification of the exact file and migration path. This ability to verify the data independently brings us back to the principles of distributed security discussed (Anderson, 2001)and brings transparency across the cloud platforms. However, one of the challenges associated with using the Ethereum blockchain is the gas fee required for each transaction. The gas fees cover the computational cost of processing and validation of transaction on the blockchain. Figure 5.2 shows an example of a gas fee incurred during the metadata recording process. In this instance, the transaction costs get to be important to evaluate the impact of scalability of the system. However, the blockchain benefits such as immutability and decentralization provide strong reasons for these costs particularly for applications requiring high data integrity (Buterin, 2013) using the Ethereum blockchain makes sure that every operation has a gas cost which is calculated as

$$\text{Total Transaction cost} = \text{Gas Used } x \text{ Gas price}$$

Whereby the gas prices vary based on network demand as the transaction costs can fluctuate significantly High gas fees, especially during high network activity, could impact the scalability, making frequent data migration events expensive in large-scale applications to minimize financial outlay during development, this system was initially tested on the Sepolia test network, which allows transactions without incurring real gas costs. using a test network provided a realistic practical environment to validate the proper functionality and cost-effectiveness of the migration protocol without financial risk. This development approach makes up a low-cost proof of concept that can be changed before deployment on a mainnet. However, for large-scale, production-level deployment, several strategies can improve and reduce transaction costs while maintaining the advantages of blockchain like transaction batching techniques. Figure 5.2 shows the gas fee incurred on the Sepolia test network, illustrating a typical transaction cost in the metadata recording process. Although no monetary costs are incurred on Sepolia, this representation shows the economic considerations for mainnet deployment.

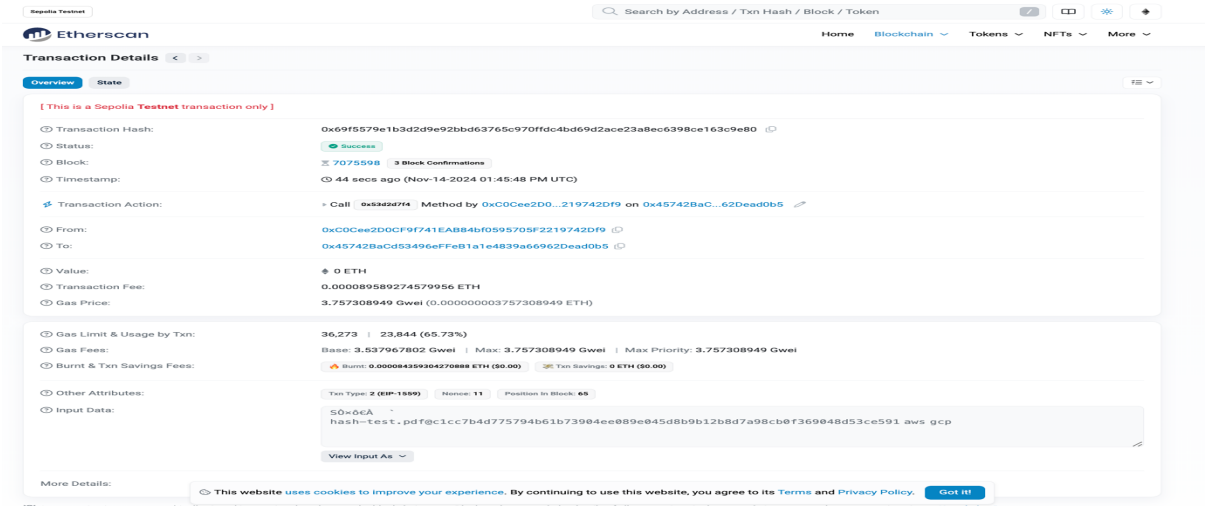

Fig 5.1 - Etherscan Display of migration transaction on blockchain

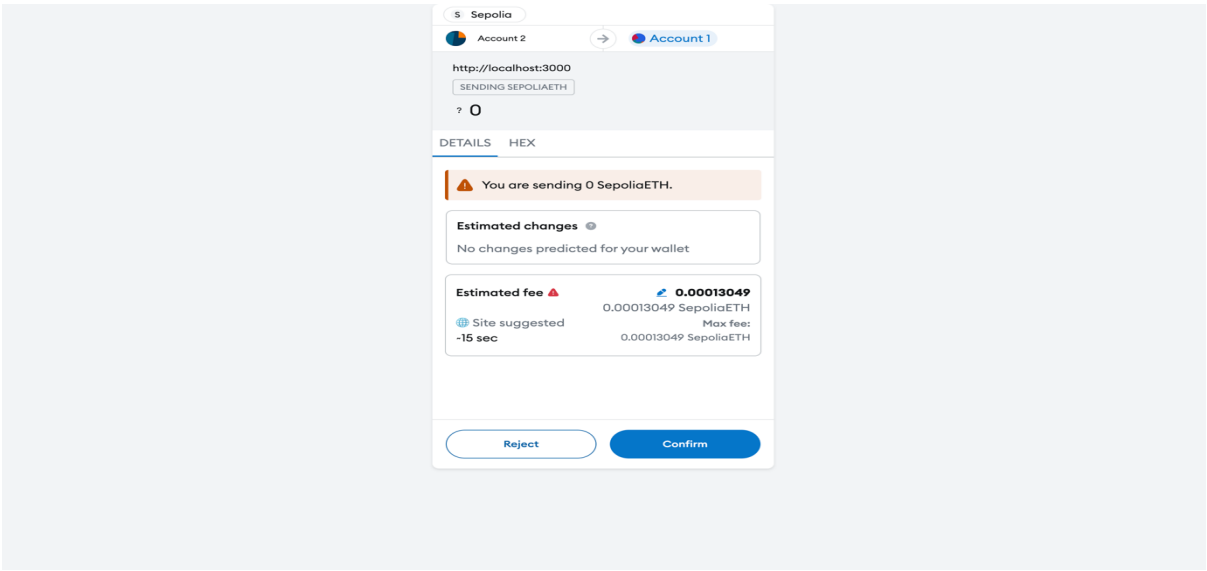

Fig 5.2 - Gas Fee and Transaction Cost for Blockchain Metadata Storage

6.2 Performance Evaluation

6.2.1 File upload performance evaluation

The performance evaluation of the file uploads across the different cloud service providers involved the measurement of upload times for various file sizes ranging from 10 MB to 1 GB. These measurements are important for understanding how the underlying cloud infrastructure and data processing methods affect the latency of the upload. As talked about by (Armbrust, et al., 2010) network latency and resource scalability are important factors influencing performance in cloud systems. The differences in the architecture of AWS, Azure, and GCP lead to variations in the upload times, which were consistently recorded and plotted.

During the evaluation, the encryption of files before upload added computational overhead, leading to longer upload times for larger files. This encryption process, which utilizes AES-256 encryption, ensured the data confidentiality and aligned with industry standards for secure cloud computing, as detailed by (Buyya, et al., 2013) For smaller files, the encryption overhead was relatively small compared to the overall upload time. However, as the file sizes increased, the proportion of time spent encrypting the data became more significant, which impacted the upload metrics.

The recorded upload times also showed the impact of cloud providers specific optimizations, such as faster storage systems and optimized network bandwidth. For instance, in figure 6 AWS showed consistent upload speeds across different file sizes, while Azure exhibited slightly higher upload times for larger files, potentially due to its handling of encryption metadata. GCP, known for its advanced data processing infrastructure, showcased a balance between performance and cost efficiency, as described by (Buyya, et al., 2013)

These findings showed that while encryption and file size influence upload times, the choice of the cloud providers themselves plays an important role in achieving the best performance. This evaluation really shows the importance of understanding the infrastructure disparities.

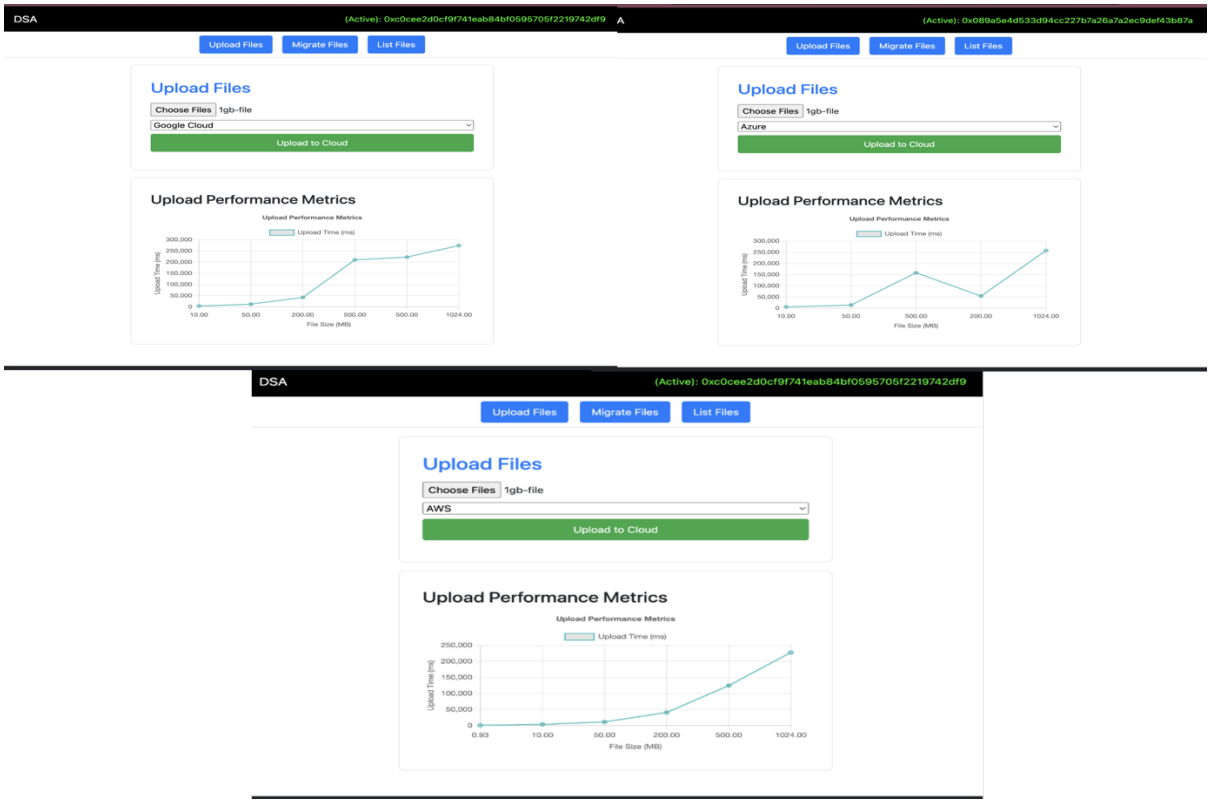

Fig 6 – Upload Metrics for the cloud service providers

6.2.2 File migration performance evaluation

The migration of files between cloud providers involved downloading and decrypting a file from the source provider, re-encrypting it, and uploading it to the target provider. The recorded

metrics included the (Li, et al., 2012)download, verification, encryption, upload, and total migration times for identical files across AWS, Azure, and GCP. This evaluation aligns with (Li, et al., 2012)who goes deep in the role of infrastructure and task scheduling in determining the cloud performance.

The total migration time varied across providers, reflecting differences in their storage efficiency, data transfer speeds, and encryption mechanisms. For instance, in figure 6.1, the migration from AWS to Azure recorded a total time of 3557 ms, with download and upload times contributing significantly. Azure to GCP migration showed a total time of 2970 ms, demonstrating Azure's relatively slower upload performance and GCP's efficient data ingestion processes. Finally, GCP to AWS migration exhibited the shortest total time of 2568 ms, highlighting AWS's optimised infrastructure for data uploads.

The evaluation also revealed the computational cost of cryptographic operations during migration. While encryption and decryption added a modest delay, their contribution to maintaining data confidentiality and integrity was justified. This is consistent with (Buyya, et al., 2013), who emphasized the need for secure data handling during inter-cloud operations.

Overall, these results demonstrate that migration latency is influenced not only by the file size and encryption requirements but also by the unique characteristics of each CSP. This evaluation provides critical insights into the challenges and opportunities of developing efficient multi-cloud migration systems.

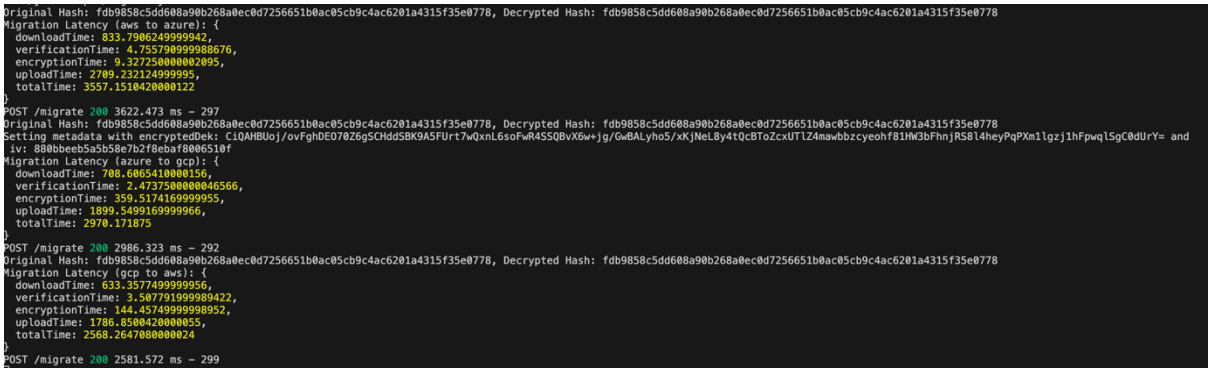

Fig 6 – Migration Metrics between the cloud service providers

In all the metrics observations, a significant observation made was that files uploaded directly via cloud providers' native consoles—such as AWS, Azure, or GCP—could not be migrated through the framework. This result shows a deliberate security feature embedded within the system. When files are uploaded directly to these consoles, they bypass the encryption and metadata protocols enforced by my application. These files lack critical metadata, including the encrypted Data Encryption Key, Initialization Vector and original file hash, which are essential for ensuring secure and trackable migrations.

During the migration process, the framework performs an integrity check by comparing the file’s hash against its associated metadata. This verification ensures the file has not been

tampered with or altered. Files uploaded outside the application's encryption framework inherently fail this verification, as they do not contain the expected encryption or metadata. This mechanism effectively blocks the migration of files not processed through the application's secure protocols.

This restriction is a key security feature, preventing unauthorized or unverified files from entering the migration pipeline. By enforcing encryption and metadata integrity as prerequisites, the system mitigates the risks of data corruption, unauthorized access, and potential leakage during migration which improves the trust in the migration process.

6.3 Comparison table

Feature	This study	(Witanto, et al., 2023)	(Kanagachalam, et al., 2022)	(Khan, et al., 2023)	(Tošić, et al., 2023)
Cross-Cloud Interoperability	yes	no	no	no	no
Real-Time Verification	yes	no	no	no	no
End-to-End Encryption	yes	partial	no	partial	no
Comprehensive Metadata Tracking	yes	no	no	no	no
Cloud-to-Decentralized Compatibility	yes	no	no	yes	no

The comparison table highlights the different contributions of the current study in the field of multi-cloud data migration and interoperability compared to prior research efforts. It evaluates the features provided by this study against the works of (Witanto, et al., 2023) (Kanagachalam, et al., 2022)), (Khan, et al., 2023)), and (Tošić, et al., 2023)

The table shows that this study uniquely delivers cross-cloud interoperability, real-time verification, end-to-end encryption, comprehensive metadata tracking, and cloud-to-decentralized compatibility. Unlike the prior studies, which often focus on isolated aspects such as encryption or metadata tracking, this study provides a holistic approach by integrating these features into a unified framework. For example, (Witanto, et al., 2023)fail to address cross-cloud interoperability, and none of the other studies incorporate real-time verification, which is important for maintaining data integrity during migration. These gaps underline the limitations of earlier approaches in addressing the practical challenges faced by organizations adopting multi-cloud strategies. End-to-end encryption is partially covered in prior works, such as those by (Kanagachalam, et al., 2022) and (Khan, et al., 2023) However, these solutions lack wide coverage, often focusing on encryption standards without addressing inter-provider compatibility or real-time checks. Comprehensive metadata tracking is entirely absent in prior studies, indicating a significant gap in ensuring data transparency and traceability—a feature critical for industries with stringent compliance requirements like healthcare and finance. Furthermore, cloud-to-decentralized compatibility, a unique feature of this study, enables seamless integration with blockchain-based decentralized networks. This approach extends

beyond traditional cloud systems, offering organizations the ability to securely verify data integrity using blockchain's immutable ledger, a crucial advantage over traditional methodologies. In conclusion, the table underscores the wide nature of this study in addressing multi-cloud challenges compared to other solutions in prior research. It focuses on the importance of combining interoperability, real-time verification, and security to set new benchmarks for multi-cloud data migration.

6.4 Discussion

The research shows important strides in focusing and addressing the challenges of multi cloud data migration, particularly through secure and interoperable blockchain protocols. The results validate the ability of the framework to achieve cross-cloud interoperability, real-time data verification and end to end encryption. However, several limitations of the framework gives opportunity for further discussion. One notable limitation is the reliance on static credentials, which restricts the framework to a single administrator account. While this approach simplifies the development, it is unsuitable for multiuser or enterprise environments. Dynamic credential management could be introduced to allow the system to support different accounts, allowing a broader usability for organizations with diverse user needs, furthermore the absence of dynamic user management impacts the flexibility making it less viable for large scale deployments or different collaborative use cases. Another area which would require improvement is the scalability of the framework. Current testing scenario focus on medium scale datasets, but enterprises often deal with significantly larger volumes of data which makes the framework be able to integrate distributed processing techniques like Kubernetes for container orchestration or taking advantage of serverless architectures which could make the system capable of handling bigger and more complex workloads. Lastly while the framework is technically wide, its practical application during web development remains limited and improving the framework with API endpoints and SDKs could allow developers to use it in their web and application pipelines, making rise to allow automation of data migration and verification process. These improvements could broaden its appeal and make it a valuable tool for developers. In Summary, while the research effectively addressing its primary goals, these limitation and areas of improvement provide clear directions for refinement.

x

7 Conclusion and Future Work

This research set out to answer the question: **How can interoperable blockchain protocols facilitate secure and efficient data migration across multi-cloud environments while maintaining encryption integrity and compliance?** The objectives were to analyze challenges in multi-cloud data migration, develop a blockchain-based protocol to address these challenges, evaluate the performance of the proposed solution, and identify the opportunities for scalability and future extensions. The framework developed in this research successfully talks about the important challenges of cross-cloud data migration, including real-time verification, consistency of encryption and communication between the cloud providers and takes advantage of blockchains decentralized nature to provide an unchangeable record of migration activities, ensuring there is trust and data integrity during the transfers. Testing validated the framework's efficacy in maintaining encryption standards, ensuring real-time verification, and achieving seamless integration across multi-cloud environments however the limitations and implications as discussed in chapter 6.4 are far reaching, particularly in

industries, where secure data migration is important. The ability to maintain the compliance with the regulatory standards such as HIPAA and GDPR while yet still making the operational efficiency is met makes this approach a valuable addition in this field of cloud computing Future work could build on these foundations by looking at the limitations discussed and looking also at additional applications. First, the implementation of dynamic credential and identity management using federated systems like OAuth or Single Sign-On (SSO) would make the system accessible to multiple users and enterprise level operations. Second, introducing modular APIs and developer tools that would allow easy integration into development pipelines. Beyond the technical improvements, a follow up research project could focus on evaluating the framework in real world scenarios such as large scale healthcare networks or financial institutions , where the system's ability to handle regulatory compliance and operational complexity can be stress tested and also exploring new integrations with new technologies like quantum safe encryption or multi-party computation could extend the frameworks importance in growing cloud ecosystems. Then from a commercial perspective , this framework could serve as the backbone for a SaaS product tailored for managing cloud environments with the features of real-time compliance auditing, migration analytics dashboard and support for decentralized storage systems like IPFS could make it a compelling solution for businesses navigating the cloud infrastructures which could be a bit complex and a future research direction could explore partnerships with cloud services providers to standardize the interoperability protocols which will end up reducing barriers to secure multi cloud adoption. In conclusion, this research demonstrates the potential of blockchain-based protocols overcoming multi cloud data migration challenges and while the framework is successful in addressing its main objectives, and bringing the principle of "make it work, then make it better" into play to guide refinements. Expanding scalability, usability, and flexibility will unlock the potential for this research to serve as a benchmark for secure multi-cloud data management, offering practical solutions for real-world challenges meaningful future work lies in scaling the system for wider adoption and look at new innovative use cases to further improve its impact in cloud computing ecosystems.

8 References

Anderson, R. J., 2001. *Security Engineering: A Guide to Building Dependable Distributed Systems.* 2nd Edition ed. New York: John Wiley & Sons.

Armbrust, M. et al., 2010. A View of Cloud Computing. *Communications of the ACM,* 54(4), pp. 50-58.

Barker, E., 2020. *Recommendation for Key Management: Part 1 – General,* s.l.: NIST special Publication.

Buterin, V., 2013. *Ethereum: A Next-Generation Cryptocurrency and Decentralized Application Platform,* s.l.: Ethereum .

Buyya, R., Vecchiola, C. & Selvi, S. T., 2013. *Mastering Cloud Computing Foundations and Applications Programming.* s.l.:Science Direct.

Daemen, J. & Rijmen, V., 2002. *The Design of Rijndael AES - The Advanced Encryption Standard.* 1st Edition ed. s.l.:Springer Nature.

Damgård, I. B., 2001. *A Design Principle for Hash Function.* s.l., Springer Nature.

Dinh, T. T. A. et al., 2018. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEE transactions on knowledge and Data Engineeering,* 30(7).

Flexera, 2024. *Flexera.* [Online]
Available at: https://info.flexera.com/CM-REPORT-State-of-the-Cloud?lead_source=Organic%20Search
[Accessed 2024 November 2024].

Kanagachalam, S., Tulkinbekov, K. & Kim, D.-H., 2022. BloSM: Blockchain-Based Service Migration for Connected Cars in Embedded Edge Environment. *Electronics,* 11(3).

Katie Costello, M. R., 2021. *Gartner.* [Online]
Available at: https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-23-percent-in-2021
[Accessed 19 November 2024].

Kavis, M. J., 2014. *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS).* 1st Edition ed. s.l.:Wiley.
Khan, H., Zahoor, E., Akhtar, S. & Perrin, O., 2023. A Blockchain-Based Approach for Secure Data Migration From the Cloud to the Decentralized Storage Systems.. *International Journal of Web Services Research,* 19(1), pp. 1-20.
Li, J. et al., 2012. Online optimization for scheduling preemptable tasks on IaaS cloud systems. *Elsevier,* Volume 72, pp. 666-677.
MarketsandMarkets, 2021. *MarketsandMarkets.* [Online]
Available at: https://www.marketsandmarkets.com/Market-Reports/blockchain-technology-market-90100890.html
[Accessed 21 November 2024].
Mell, P. & Grance, T., 2011. *The NIST Definition of Cloud Computing,* s.l.: National Institute of Standards and Technology.
Nakamoto, S., 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System.,* s.l.: metzdowd.com.
O'Hara, B. T. & Ben Malisow, 2016. *Certified Cloud Security Professional (CCSP) Official Study Guide.* s.l., Sybex.
Parliament, E., 2016. *EU Regulation.* [Online]
Available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj
[Accessed 21 November 2024].
Schneier, B., Ferguson, N. & Kohno , T., 2011. *Cryptography Engineering: Design Principles and Practical Applications.* 1st Edition ed. n.p.:John Wiley & Sons.
Tošić, A., Vičič, J., Burnard , M. & Mrissa , M., 2023. *A Blockchain Protocol for Real-Time Application Migration on the Edge,* Slovenia: Multidisciplinary Digital Publishing Institute.
Verizon, 2021. Verizon 2021 data breach investigations report. [Online]
Available at: https://www.verizon.com/about/news/verizon-2021-data-breach-investigations-report
[Accessed 21 November 2024].
Winkler, V. J., 2011. *Securing the cloud.* 1st Edition ed. s.l.:Syngress.
Witanto, E. N., Brian , S. & Lee, . S.-G., 2023. *Distributed data Integrity verification scheme in multi-cloud environment,* Basel: PubMed.
Zhang, Q., Cheng, L. & Boutaba, . R., 2010. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications,* 1(1), pp. 7-18.