

DEMEDIA – DECENTRALIZED SOCIAL MEDIA PROTOCOL

Dhananjani G.G.S.

IT20137496

B.Sc. (Hons) Degree in Information Technology
Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

MECHANISM FOR DATA INTEGRITY PRESERVATION

Dhananjani G.G.S.

IT20137496

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of
Science specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

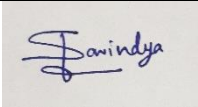
September 202

DECLARATION

I declare that this is our own work, and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

IT20137496	Dhananjani G.G.S.	
-------------------	--------------------------	---

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor:

Date:

ABSTRACT

The decentralized social media protocol is a modern technology that is aimed to overcome the issues of centralized social media platforms, including censorship and data privacy concerns. Through this protocol, users are allowed direct ownership and control over their data, ensuring a more transparent and user-centric experience. It enables users to communicate and interact with others without involving a central or single source. Our research will involve developing a protocol called "DeMedia" that aims to resolve the limitations and constraints associated with centralized social media platforms. The DeMedia protocol is implemented to offer users enhanced control over their data and ensure their privacy. Through the development of this protocol, we aimed to address the concerns of data privacy, security, and ownership, while providing users with a more private and safe social media experience. The Data Integrity component of our research primarily focused on safeguarding the authenticity and consistency of user data. It emphasizes the importance of maintaining the uniqueness of information while ensuring that it remains consistent across all platforms.

This component was designed to implement a cryptographic system which facilitates the secure storage of authenticated user data directly on the user's device. This mechanism improves the security and reliability of the user's data by ensuring that it remains tamper-proof and unaltered. And develops an effective and efficient system that maintains user privacy and control while enhancing data integrity. Building user trust requires ensuring the legitimacy and authenticity of the data given on the platform. The data integrity component utilizes cryptographic techniques such as hashing mechanism and data encryption mechanisms to secure the data and prevent unauthorized modifications or access. Ultimately, the decentralized social media protocol and its data integrity component provide a hopeful solution for developing a more transparent and secure social media platform. With the increasing of data privacy and censorship of user data, this technology could revolutionize the way people communicate and share information online. This could lead users to feel more confident and secure in expressing themselves and sharing information.

Keywords: *Data Integrity, Censorship, Privacy, Security, Authenticity, Consistency, Tamper-proof, Cryptographic Techniques, Hashing Mechanism, Data Encryption*

ACKNOWLEDGEMENT

The success of this research is due to the efforts of my supervisor, Mr. Kavinga Yapa Abeywardena, as well as the rest of the team. Additionally, I'd like to express my gratitude to the Department of Information Technology at the Sri Lanka Institute of Information Technology, as well as to CDAP academics and staff, for enabling this research.

TABLE OF CONTENT

DECLARATION.....	i
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENT	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
1.0 INTRODUCTION.....	1
1.1 Background And Literature	5
1.2 Research Gap.....	9
1.3 Research Problem	11
1.4 Research Objective.....	13
1.4.1 Main Objective	13
1.4.2 Specific Objectives	13
2. METHODOLOGY.....	15
2.1 Requirement Gathering.....	15
2.1.1 Past Research Analysis	15
2.1.2 Refer Official Documentations.....	15
2.1.3 Identify Existing Methodologies	15
2.2 Feasibility Study	16
2.2.1 Technical Feasibility	16

2.2.2	Schedule Feasibility.....	16
2.2.3	Economic Feasibility	16
2.3	Requirement Analysis.....	17
2.4	Software Development Life Cycle (SDLC)	17
2.5	Proposed System Design.....	19
2.5.1	System Overview Diagram (Overall)	19
2.5.2	System Overview Diagram (Individual)	21
2.6	Commercialization	22
2.7	Implementation	24
2.7.1	Background Establishment For Implementation.....	24
2.7.1.1	Programming Language Selection: Go Lang	24
2.7.1.2	Database System: PostgreSQL	24
2.7.1.3	Collaboration and Version Control: GitHub	25
2.7.1.4	Cloud Deployment: Amazon Web Services (AWS)	26
2.7.1.4.1	Deployment Using Amazon EC2 Instances	26
2.7.1.5	Containerization: Docker	27
2.7.1.5.1	Deployment as Docker Containers	27
2.7.2	Setting Up CI/CD Pipeline	29
2.7.2.1	Continuous Integration (CI) Pipeline.....	29
2.7.2.2	Continuous Deployment (CD) Pipeline	29
2.7.3	Development	32

2.8 Testing	35
2.8.1 Unit Testing	35
2.8.2 Continuous Integration (CI) and Deployment (CD)	35
2.8.3 Smoke Testing	36
2.8.4 Validation using practical implementations.	37
2.8.4.1 Benchmarking with Infrastructure as Code (IaaS) Approach....	37
3. Results and Discussions	39
3.1 Results	39
3.1.1 Demo Social Media Platform Results	39
3.1.2 Benchmarking Results	41
3.1.3 Results from overall practical implementations	41
3.2 Research Findings	42
3.3 Discussion	43
4. CONTRIBUTION SUMMARY	44
5. CONCLUSION.....	45
REFERENCES.....	46
GLOSSARY.....	48
APPENDICES	50

LIST OF FIGURES

Figure 1.0.1: Centralized vs Decentralized	13
Figure 1.1.1: Hashing Algorithm	17
Figure 1.1.2: Digital Signature Mechanism	18
Figure 1.1.3: Cipher Block Chaining Encryption	19
Figure 1.0.1: Figure 2.4.1: Software Development Life Cycle	29
Figure 1.0.1: Figure 4.4.1.1: System Overview Diagram	30
Figure 2.5.2.1: Proposed Diagram for Data Integrity	32
Figure 2.7.3.1: Code snippet of data integrity verification	44
Figure 2.7.3.2: Stored signatures on users database	45
Figure 3.1.2.1: Screen capture of data integrity on demo social media platform	51
Figure 3.1.2.2: The panel of Decentralized Data integrity	52
Figure 3.1.3.1: Output from benchmark application	55

LIST OF TABLES

Table 1.2.1: Comparison of existing decentralized social media platforms	20
Table 2.7.1.4.1.1: Specifications of EC2 instance	37

LIST OF ABBREVIATIONS

IPFS - InterPlanetary File System

DeSo - Decentralized Social

CBC - Cipher Block Chaining

1.0 INTRODUCTION

Connectivity and communication have taken on new dimensions in the digital era of today. A Social media platform is a website or application that enables users to create and share information, communicate with others, and take part in social networking. Typically, social media platforms provide their users with the tools and capabilities they need to create and publish various types of content including text, images, videos, and audio. [1] These platforms have grown in popularity, and billions of people use them to communicate with friends and family, share information and news, and interact with community groups and cultures from across the globe. [2]

Social media platforms can be broadly categorized into two types as centralized and decentralized. Currently, the majority of social media platforms are centralized, which means that they are owned and operated by a single entity and have a central server that manages all user data and content. And the platform owner has full control over the platform's features, policies, and access to user data. [3] Centralized social media platforms are convenient and easy to use, but they also present some risks in terms of data privacy and security. Users must trust the platform owner to manage their data responsibly, and there is always the risk of data breaches or abuse by the platform owner or their partners. [4]

In contrast to this centralized model, there is an increasing concern in platforms that emphasize user autonomy and data protection. Decentralized social media platforms are designed to distribute the control and ownership of the platform among its users, rather than being owned and operated by a single entity. Users have direct control over their data and content, which can help to strengthen privacy and security. User data and content are stored on a distributed network of computers. [5] Decentralized social media protocols use various mechanisms to ensure data integrity, including cryptographic hashing and encryption methods. By dispersing power and ownership among its users and ensuring data integrity, decentralized social media platforms can provide their users more control over their privacy, security, and decision-making. [6] This approach empowers individuals and reduces reliance on a single authority.

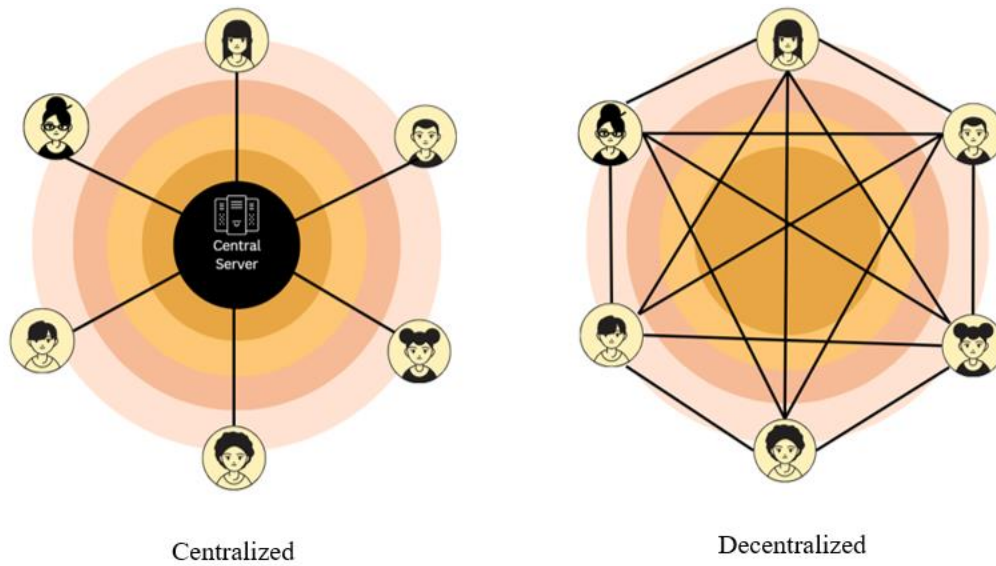


Figure 1.0.1: Centralized vs Decentralized

“DeMedia” is a decentralized social media protocol that allows anyone to build their own decentralized application using its data layer and caching mechanism. With DeMedia, even nontechnical users can easily deploy peers on their devices with low operational cost. To ensure the integrity of user data, DeMedia uses a cryptographic mechanism that publishes the hash of set of hashes on IPFS network. [7] This mechanism helps to ensure that user data cannot be tampered with or altered by anyone. It provides greater security and trust in the platform.

In the context of any application, data integrity means that the data keeps unmodified and unaltered from its original state, unless a change is expected as a result of a specific or authorized action. In any application, data integrity serves a fundamental component. [8] It ensures that information remains accurate and consistent which improves trust among users. And it is one of the main aspect of the decentralized social media protocol. Because in a decentralized environment, user data and information are stored on a distributed network of computers rather than a central server managed by the platform owner. The integrity of this data must be guaranteed to be unaltered and untampered, as well as accurate and consistent. [9]

Without data integrity, users could not have confidence in the platform's ability to preserve their data and might be afraid to utilize the platform. By ensuring data integrity, decentralized social media platforms can provide a secure and trustworthy environment for users to share and communicate without having to worry about the security of their data. Moreover, due to the absence of a single point of failure in decentralized systems, hackers have to face increased challenges; they are unable to simply breach one main hub to access or manipulate any and all data.

Currently, there are various mechanisms available to maintain the integrity of data in decentralized application. Therefore, as my research component, I have implemented a proper data integrity mechanism on top of IPFS. DeMedia, which is a decentralized social media protocol that stores user data in the user's device, letting them have full control over it and making changes as their discretion. As a result, there is less chance of data corruption or tampering by outside parties and users have more control over their data. [10] Data integrity is a comprehensive process that extends beyond simply preserving information. It requires a commitment to safeguarding user data accuracy and consistency through time and across different platforms or systems. Decentralized social media platforms can secure user data even in the absence of a centralized authority by implementing data integrity methods including cryptographic hashing and data encryption.

It enables a way to validate the legitimacy of user data and content by implementing hashing algorithms as a cryptographic mechanism to establish a hash value for each piece of user-generated content or data. [11] Because each piece of data has a unique hash value. A different hash value will be generated if the data is modified or tampered with in any way by someone. The platform is able to identify any attempts to tamper with the user data by comparing the original hash value to the recently generated hash value.

There are two approaches to measure the data integrity of decentralized social media protocols such as quantitatively and qualitatively. In quantitative measurements, the level of accuracy, consistency, and reliability of the data can often be evaluated numerically. [12] Quantitative measures involve evaluating the redundancy and

consistency of the data stored on the IPFS network, which provides a distributed file system. In Qualitative measures, it is possible to check the hash value or compare the encrypted data with the initial data to make sure it has not been modified. It implies non-numerical measurement of the data's reliability, credibility, and validity. Both quantitative and qualitative measures are important for evaluating the data integrity of decentralized social media protocols. [13]

The importance of data integrity in decentralized social media protocols cannot be overwhelmed in the developing world of digital communication. The primary focus of the component is the implementation of mechanisms addressed to ensure data integrity within a decentralized network. As decentralized systems gain popularity among users, it becomes increasingly important to guarantee the authenticity and consistency of data distributed across multiple nodes. This seeks to provide a comprehensive understanding of how data integrity can be maintained in an environment without centralized control.

1.1 Background And Literature

Data Integrity is a main aspect of both centralized and decentralized social media platforms. When user data is tampered with, incorrect information can propagate. This not only misleads individuals, but also affects their privacy and security. Data protection is critical for maintaining trust as well as security in digital platforms. For instance, tampering with a user's private messages or personal information might have huge consequences for the user's privacy and security. [8] Therefore, data integrity has received increasing attention in numerous research studies. This increase in research not only emphasizes the significance of data integrity in present-day society, but it also aims to develop reliable strategies to safeguard it from potential risks.

In a decentralized social media platform, user data and content are stored on user's device and a distributed network such as IPFS network instead of a central server. [7] Hence, maintaining the accuracy, integrity, and security of user data requires assuring data integrity. And establishing trustworthiness among users of a decentralized social media site requires data integrity. Users may be unwilling to utilize the platform or share any of their personal information if they have no confidence that their information is protected. It is essential for platforms to safeguard their data while also making sure users know about these security measures.

The amount of data that has to be saved is increasing fast in the big data era. As a result, due to its accessibility and huge capacity, cloud storage has become increasingly popular to store information storage. To reduce expenses, certain cloud service providers, however, could remove information that is not usually accessed, which could lead to users losing important data. A new technique based on blockchain technology has been recommended for validating the integrity of recoverable information saved to the cloud storage in order to reduce issues about data security and privacy. [14]

There are several benefits to using blockchain technology to preserve the integrity of information in decentralized social media networks, and there are some possible drawbacks as well. [15] One of the main challenges with using blockchain for data

integrity is scalability. A network of nodes that process blockchain transactions must decide on each transaction's legitimacy. As more operations are added to the blockchain network, it may become slower and less effective. And that leads to delays and blocks. Decentralized social media networks that produce a lot of information and content could struggle with this issue in particular. [16]

DeMedia is not a blockchain-based decentralized social media network due to these difficulties with the blockchain. For address these concerns in integrity of user data, uses a method where a hash value of a user's data is created and encoded, and then stored on the user's device. And then a set of these hash values are also stored on the IPFS network. [7] IPFS is a distributed system that allows for the storage of immutable data, eliminates redundancy. Moreover, it offers address information for storage nodes to make it easier to discover files on the network. [17] When it comes to decentralized user data storage, IPFS has several advantages, including increased security, performance, and reliability. [18] When a user wants to retrieve their data, DeMedia decrypts the hash value to ensure that the data hasn't been altered. In this manner, the data's integrity is ensured, and tampering is prevented.

DeMedia stores user's data as a hash value. The users' data can be hashed to generate a fixed-length digest that can then be signed with the user's private key. [19] As any modifications to the data will result in a different digest value and this assures that the data has not been altered.

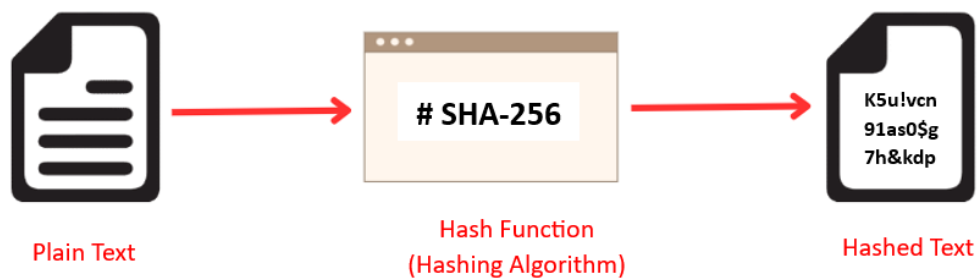


Figure 1.1.1: Hashing Algorithm

After generating hash value for user data, it can be signed using a digital signature. Digital signature is a cryptographic method that offers a way to determine the authenticity of messages. It ensures that the message is authentic and came from the specified source. The sender signed the message using the private key and the receiver can verify the message using sender's public key. Decentralized social media protocols can offer a secure and trustworthy platform for communication and data sharing by using digital signatures. The use of this method not only increases user trust in the platform, but it also discourages potential malicious individuals from interfering with the data. As more people become aware of the importance of digital security, platforms that prioritize such safeguards are likely to gain popularity and trust.

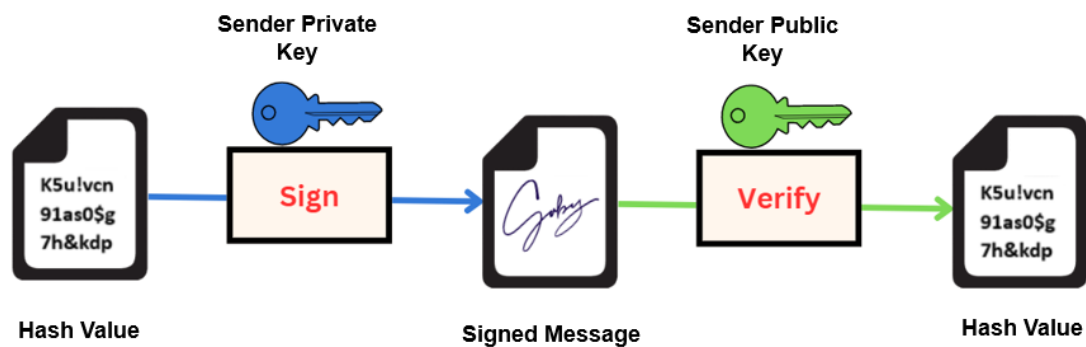


Figure 1.1.2: Digital Signature Mechanism

The digitally signed data is saved in the user's device. Then get the set of signed hash values which is saved in the user's device and generate a single hash value for the set of hashed data. The set of hash values can be a hundred or thousand hash values. After that, the hashed set of signed value is stored in the IPFS distributed network. To store in IPFS, the hashed value is split into smaller chunks using Cipher Block Chaining (CBC) encryption method, that is a mechanism for ensuring the integrity and confidentiality of the data. The hashed value is divided into blocks of a fixed size. When retrieving the data, storing the hash values on the IPFS network, and comparing them to the computed hash values ensure that the data has not been altered or

corrupted. This method provides an extra layer of security, ensuring users can trust the authenticity of the data they access. Decentralized platforms may further enhance their popularity as reliable as well as secure digital alternatives by taking advantage of these kinds of mechanisms.

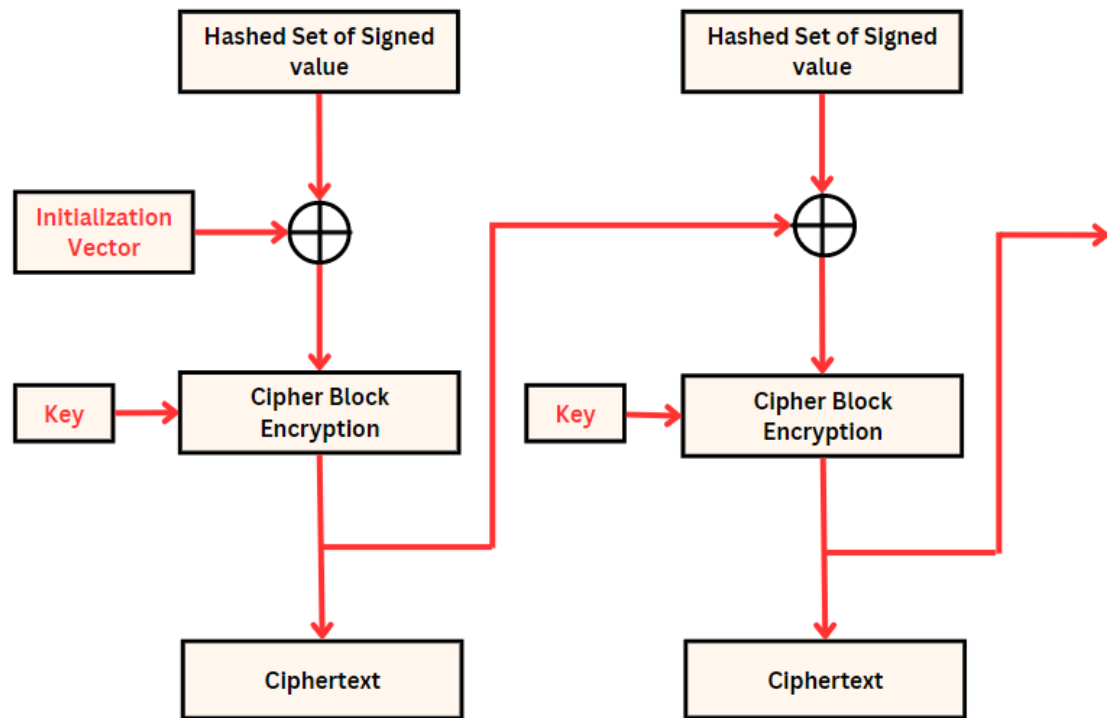


Figure 1.1.3: Cipher Block Chaining Encryption

1.2 Research Gap

The primary research gap in this research aims to fill the gap that exists in current methodologies in data integrity of decentralized social media platforms.

	Mastodon	DeSo	DeMedia
Generate hash for user data	Yes	Yes	Yes
Sign user data	Yes	Yes	Yes
Keep data Integrity in User's device	No	No	Yes
Use IPFS for keep Integrity	No	No	Yes

Table 1.2.1: Comparison of existing decentralized social media platforms

Mastodon is a popular decentralized social media platform which uses blockchain mechanism for store data. It generates hash values for user data and signs using digital signature. But it does not keep data with user device and does not use IPFS distributed network for store hashed and signed user data.

DeSo is a currently available decentralized social media platform. It uses first layer blockchain to build and scale storage-heavy applications to billions of users around the world. [20] Therefore, it does not keep data in user's device. By utilizing blockchain technology and decentralized infrastructure, DeSo is focused on providing a social media experience that is more transparent, reliable, and safe data processing and user centric. [21] It does not use IPFS network for store data.

Although Mastodon and DeSo are decentralized social media platforms, they have some issues because of using blockchain technology. There are some considerable drawbacks to blockchains. [22] Such as slower processing, harder to scaling, high power consumption, high cost and store immutable data. Therefore, it is not appropriate to use blockchain technology for a decentralized social media network.

However, although a lot of research has been done about data integrity of decentralized social media platforms, no research has been done to provide a data integrity mechanism for stored user hash data in user's device and stored set of signed hashes in IPFS distributed network. Therefore, this research aims to explore the feasibility and effectiveness of using a decentralized approach to maintain data integrity of both stored user data in the user's device and IPFS distributed network.

1.3 Research Problem

Existing centralized social media platforms need to be improved to better serve the needs of users and society. These platforms have several data integrity issues that create massive ethical, privacy, and public policy concerns. The following issues are widespread currently on social media platforms in user data integrity.

Users may have limited control over their data and platform owners have the authority to control user data. Owners of centralized social media platforms gather and store enormous quantities of personal information about their users, which is vulnerable to hacking, misuse, and abuse. Large volumes of user data are stored on a central server by centralized social media platforms. This makes them a tempting target for hackers. And Users' personal information, login passwords, and private messages may be compromised by data breaches. Centralized platforms have the authority to censor both content and users, potentially compromising the platform's data integrity. Overall, data integrity, privacy, and security are severely impacted by the centralized approach of social media networks.

Decentralized social media protocol is a solution to address the issues caused by centralization. However, before we can use this solution, there are several research problems that need to be addressed for the successful implementation of decentralized data integrity. Overcoming these challenges will allow the way for more secure and user-centric platforms.

The first challenge is about keeping user data safe and unchanged that is stored on their devices. When data is stored on user's own devices, there is a risk of it getting lost, corrupted, or tampered with. Therefore, we need a way to constantly check and make sure that the data remains as it was originally, even while it sits on countless devices around the world.

The second challenge is to identify the best mechanism for achieving the integrity of user data. Because there are many mechanisms that are already used to keep data safe in decentralized systems. Some might be highly secure but slow, while others might

be fast but not as safe. We need to sort through these possibilities to determine the appropriate balance for user data on social media platforms.

The third problem is making sure that integrity of stored user data in the IPFS network. It is essential to maintain data integrity in IPFS. Because IPFS spreads data across multiple places, there's a need to have a system that constantly checks and confirms that no piece of data has been altered without permission.

By addressing these research problems, data integrity can be verified in a decentralized social-media protocol. This indicates that as users interact, share, and communicate on these platforms, they can have confidence that their information remains authentic and unaltered. Addressing these issues not only enhances the trustworthiness of the platform but also creates the way for broader adoption, as users can be confident of the safety and accuracy of their data in a decentralized environment.

1.4 Research Objective

1.4.1 Main Objective

Implement a mechanism for decentralized data integrity preservation for user data that facilitates the development of decentralized social media protocol. This security ensures that while users interact, share, and communicate on these platforms, their personal information and shared content remain consistent and secure from unauthorized changes or breaches.

1.4.2 Specific Objectives

Ensure consistency in preparation for hashing and signing.

It is essential to maintain consistency while preparing data for hashing and signing in order to ensure the integrity of the data. This involves following a defined procedure to format and prepare the data before applying cryptographic operations such as hashing and digital signing. Consistency is essential due to even minor changes in the data might result in significantly different hash values or invalidated digital signatures. Organizations may reduce the risk of unauthorized data modifications or conflicts during these processes by maintaining consistency in how data is generated.

Ensures the data security distributed across multiple nodes.

Data distribution across multiple nodes is a fundamental principle in many distributed computing and storage systems. This approach has multiple benefits including fault tolerance, scalability, and load balancing. Distributed data security serves as essential for ensuring data availability and confidentiality in complex and distributed systems.

Verify the authenticity of the data.

It is necessary for establishing trust in digital communications and transactions by verifying the authenticity of data. Usually, digital signatures or identical cryptographic approaches are used in this process. When data is signed by a trustworthy party, receivers can verify the authenticity of the signature in order to ensure that the data hasn't been tampered with. Authentic verification ensures that the data was generated by a legitimate entity and was not altered.

Ensure that the data has not been altered or tampered with during storage.

It ensures that data remains unaltered and reliable throughout its storage. Various methods such as cryptographic hash functions and checksums are used to achieve this. Organizations can identify unwanted modifications or tampering activities by frequently comparing hash values of stored data against the initial values. In addition, storage systems for data may include redundancy and error-checking mechanisms to automatically address or recover data if any corruption or tampering is detected. Maintaining data reliability and accuracy is important especially in long-term storage or archiving systems.

2. METHODOLOGY

2.1 Requirement Gathering

The initial step for this research component involved collecting the requirements for the novel data integrity mechanism. To establish a solid foundation for the protocol, we built on the knowledge and insights of existing data integrity mechanisms. We could better define the requirements for our new protocol, ensuring it was both reliable and suitable to the challenges of modern decentralized platforms by analyzing what was already in place and understanding its strengths and weaknesses.

2.1.1 Past Research Analysis

The second phase of this research is to analyze past research of the data integrity mechanisms. The analysis of existing research papers is done to identify the problems of the existing data integrity mechanisms, latest solutions and the pros and cons of earlier methods. This allowed us to understand the research methods that are used to maintain data integrity in decentralized platforms.

2.1.2 Refer Official Documentations

Official documentation provided us up-to-date information about the technology we planned to use for our proposed system. Although previous research publications contain a large amount of information because the technologies we use are constantly updated, they may contain out-of-date information.

2.1.3 Identify Existing Methodologies

Decentralized social media platforms currently exist with a focus on maintaining data integrity. However, many of these technologies lack a comprehensive approach to enhance and improve user data security. While they may provide some level of protection against data tampering and censorship, the wide range of security concerns often remain addressed. A more comprehensive strategy is essential, encompassing strong encryption and innovative solutions to establishing privacy concerns.

2.2 Feasibility Study

2.2.1 Technical Feasibility

It is technically feasible to develop data integrity in a decentralized social media protocol that utilizes hashing algorithms, digital signatures, encoding, and data integrity mechanism in IPFS. Such an approach can significantly enhance security and trustworthiness of content shared on decentralized social media platforms. Thorough testing and optimization will be important to address the challenges and ensure successful implementation of the protocol for decentralized social media with data integrity.

2.2.2 Schedule Feasibility

The research project has been planned with the expectation that the component will be finished in approximately seven-month time frame. In the first two months we focused on gathering and analyzing the project's requirements. Then, we spent the following three months on system development. We have allocated the last two months for testing. This helped us ensure that our research objectives were feasible and aligned with our technical capabilities and the time we had anticipated.

A Gantt chart has been designed to illustrate the work that has to be completed under certain time frames. This component can be considered schedule-feasible after considering all the mentioned factors.

2.2.3 Economic Feasibility

When assessing economic feasibility, we need to evaluate the project's financial aspects. There were no budgeted tasks in the project's initial phase from the perspective of economic feasibility. Because we gathered requirements by reviewing and analyzing previous research and studies that doesn't require any additional expenses. We also choose open-source technologies and tools that are open-source. However, we spent costs during the development phase since we need to utilize cloud service providers' services.

2.3 Requirement Analysis

The requirement analysis was a fundamental phase in this research project since it helped identify several aspects that needed to be considered during the project's implementation. This step was like building a roadmap for this research. We looked at lots of different information sources to figure out exactly what we needed for the project. It helped us identify what potential challenges we might run into, what tools and technologies we could use, and how to approach the method. And it provided us with a clear idea of what our project could achieve, where its limits were, and what gaps we needed to fill in the existing research.

2.4 Software Development Life Cycle (SDLC)

The SCRUM framework, an Agile software development framework, will be used as the primary software project management framework throughout the research. The reason to choose agile methodology over other software project management methodologies such as Lean, Waterfall model, and Six Sigma is because it is best adapted for rapid and effective software development.

According to the article [23], SCRUM is a popular agile framework because it defines the systems development process as a loose collection of activities combining the finest tools and techniques a development team can devise to create a system. According to additional information in the same article [23], SCRUM implies that the system's development process is unpredictable, complex and can only be described as an overall progression.

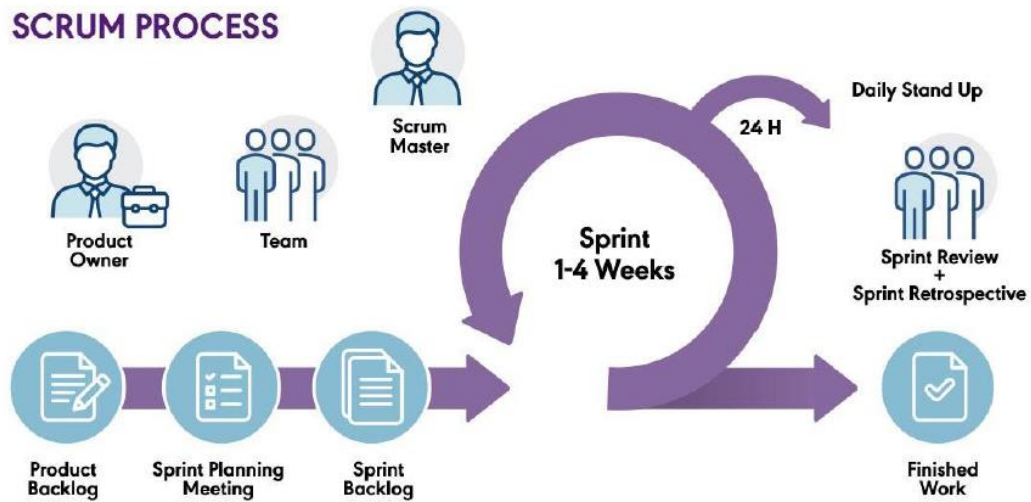


Figure 2.4.1: Software Development Life Cycle

A systematic allocation and organization of the work have been used to achieve the research's outlined objectives and achieve the desired outcomes. A detailed schedule, complete with a Gantt chart, has been made to give each part of the research sufficient time to be finished on time. In addition, the selection of appropriate technologies to effectively implement the proposed solution and demonstrate the intended results of this research has been carefully considered. As evidenced by the detailed preparation and strategic decisions made throughout this research, each step has been taken to ensure a well-structured and systematic approach to achieving the research objectives.

2.5 Proposed System Design

2.5.1 System Overview Diagram (Overall)

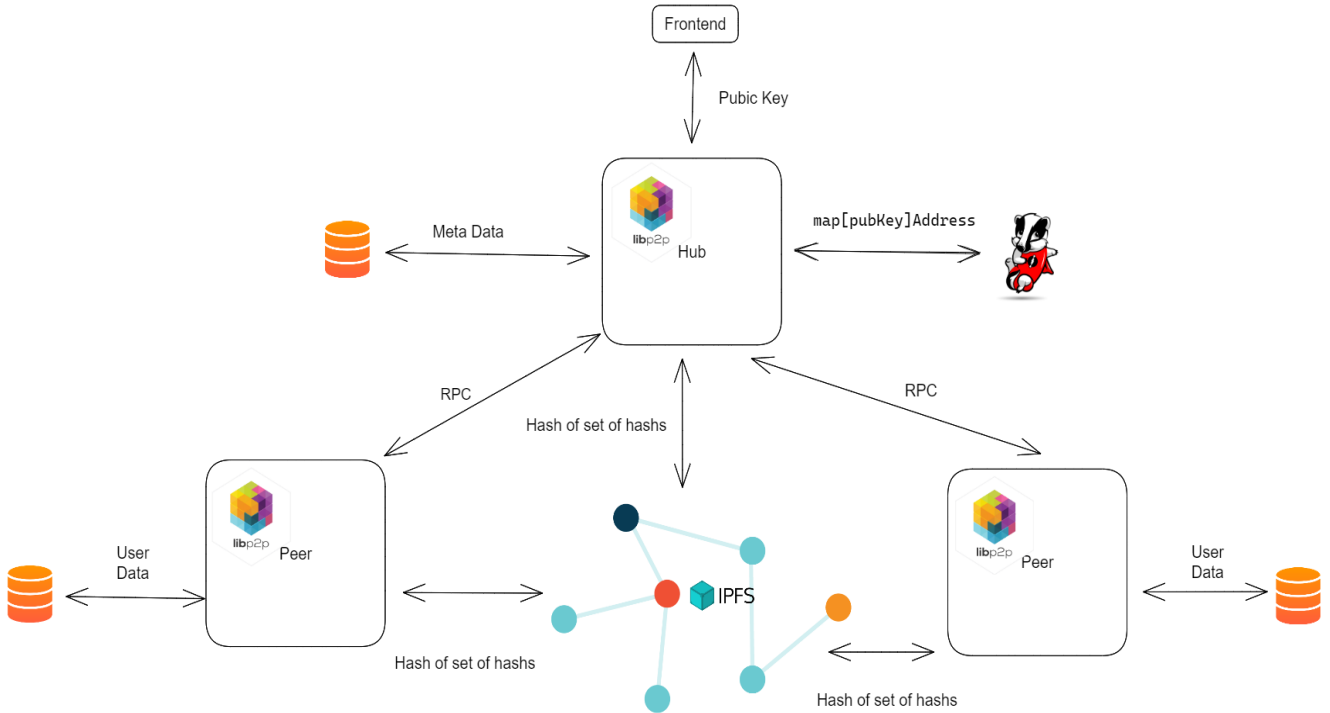


Figure 4.4.1.1: System Overview Diagram

This diagram presents an overall overview of the DeMedia architecture. DeMedia comprises four major components such as data decentralization protocol, peer-to-peer communication, decentralized data storage, and data integrity in a decentralized network. A social network platform is designed with IPFS to display these capabilities. This system infrastructure will consist of hubs, peers, and a decentralized data storage network.

The first component of the DeMedia architecture is the data decentralization protocol, which enables the platform to store user data within the users' devices rather than on centralized servers. This decentralization of data will enhance user control over their personal data, as well as increase data privacy and security. The second component of

the DeMedia architecture is peer-to-peer communication, which allows users to communicate with each other through a hub which will act only as a communicator. This will increase the speed and efficiency of communication while reducing the dependence on centralized servers. The third component of the DeMedia architecture is decentralized data storage, which will be accomplished using InterPlanetary File System (IPFS). IPFS is a protocol that enables the creation of a decentralized file-sharing network, which is more secure and fault-tolerant than centralized file-sharing networks. The fourth and final component of the DeMedia architecture is data integrity in a decentralized network. This involves ensuring that the data stored on the decentralized network is secure, reliable, and tamper-proof. This is accomplished using cryptography and other security measures to ensure that the data cannot be tampered with or compromised.

Overall, the DeMedia architecture aims to create a decentralized social media platform that offers increased privacy, security, and user control over personal data. This diagram illustrates a high-level architectural perspective of the presented social network platform.

2.5.2 System Overview Diagram (Individual)

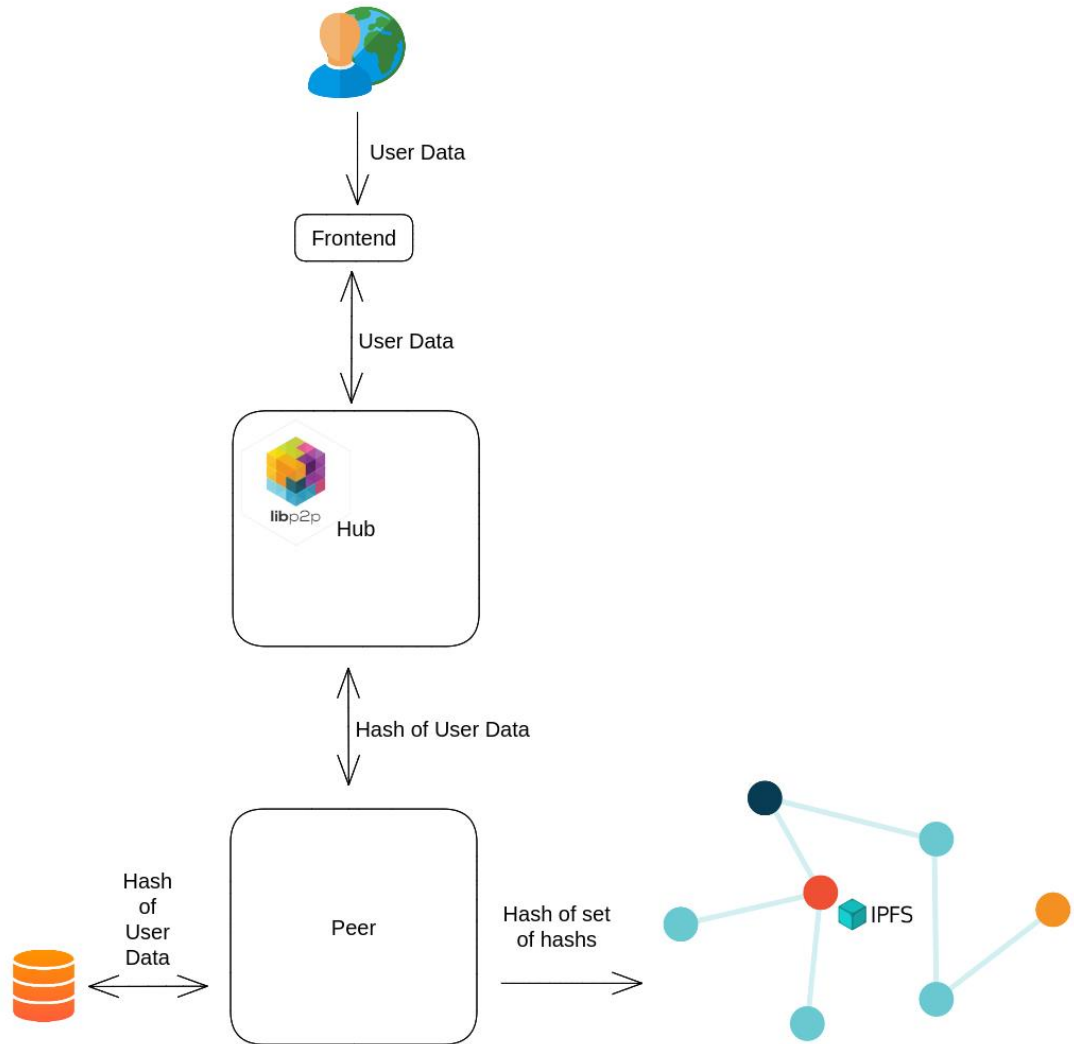


Figure 2.5.2.1: Proposed Diagram for Data Integrity

In this research component, a data integrity mechanism implemented for decentralized social media protocol. The user enters data using interface and the user data goes to hub. Then the data is hashed using the hashing algorithm. The hash of user data which is digitally signed is saved in user's device and hash of set of hash values is stored in IPFS distributed network.

2.6 Commercialization

DeMedia is focused on developing an open-source protocol that facilitates the creation of decentralized social media platforms which can be self-hosted. The project aims to provide a base model for free, which can be used by anyone interested in creating a decentralized social media platform.

In addition to the free model, DeMedia will also offer two paid models.

- Subscription-based membership model
- Advertising-based revenue model

These paid models can be governed by the host, allowing them to generate revenue from their platform. Therefore, one could describe DeMedia as a research project focused on commercializing the development of decentralized social media platforms. The project is developing a range of monetization models that can be used by hosts to generate revenue from their platforms, thereby enabling commercialization of the technology.

To further elaborate on the commercialization aspect of DeMedia, it's important to understand that the project is focused on creating a technology that can enable the development of decentralized social media platforms. By providing a free base model, DeMedia is making it easier for individuals or organizations to create their own social media platforms that are not controlled by a centralized authority. However, to sustain and grow these platforms, there needs to be a way to generate revenue. This is where the two paid models offered by DeMedia come in.

The subscription-based membership model allows hosts to charge users for access to premium features or content on their platform. This revenue can be used to cover the costs of hosting and maintaining the platform.

On the other hand, the advertising-based revenue model enables hosts to generate revenue by displaying ads on their platform. Hosts can charge advertisers to display their ads on the platform, and this revenue can be used to cover the costs of hosting and maintaining the platform, as well as generating profits.

As a summary about this proposed system, DeMedia is enabling the commercialization of decentralized social media platforms by providing a technology that allows anyone to create their own platform, along with monetization models that enable hosts to generate revenue and sustain their platforms. This could lead to a more diverse and decentralized social media ecosystem, with a greater range of platforms catering to specific sectors and communities.

2.7 Implementation

2.7.1 Background Establishment For Implementation

The team went through a careful planning and decision-making process on the way to putting the decentralized social media protocol into implementation. This phase was thought to be essential for laying a strong basis for the next development and testing phases. In-depth discussions of the decisions made regarding the project's programming language, database system, communication tools, cloud provider, and containerization technologies will be discussed in this section. Each of these choices was crucial in determining the project's ultimate outcome.

2.7.1.1 Programming Language Selection: Go Lang

After a thorough consideration, the decision was made to utilize the Go programming language, commonly known as Go Lang. There were multiple solid factors that led to this decision.

Go Lang is popular for its efficiency, simplicity, and scalability, making it an ideal choice for building decentralized systems. Its concurrency support allows multiple operations to be executed concurrently, enabling the protocol to handle a large number of users and interactions efficiently. Moreover, Go Lang offers a strong standard library and a rich ecosystem of packages, significantly expediting the development process.

The inherent support for robust error handling was considered advantageous, as it enhances the protocol's resilience and fault tolerance. In essence, Go Lang not only met the technical requirements but also aligned with the goal of creating a stable and reliable decentralized social media platform.

2.7.1.2 Database System: PostgreSQL

The selection of an appropriate database system is critical in ensuring the efficient storage and retrieval of data in any project, including the decentralized social media

protocol. PostgreSQL was chosen as the database management system, a decision influenced by several key considerations.

First and foremost, PostgreSQL is recognized as a robust open-source relational database system. Its support for complex data types and advanced indexing mechanisms makes it well-suited to handling the diverse data structures prevalent in social media applications. Additionally, strong data integrity and security features offered by PostgreSQL ensure the privacy and reliability of user data.

Furthermore, PostgreSQL's extensibility through user-defined functions and a vibrant community of contributors made it adaptable to evolving project needs. The database's ability to perform efficiently even under high loads was a crucial factor in the decision, given the typically rapid and unpredictable user activity on social media platforms.

2.7.1.3 Collaboration and Version Control: GitHub

Effective collaboration and version control are indispensable for a team working on a complex project like a decentralized social media protocol. To achieve this, GitHub was adopted as the primary platform for code collaboration and integration. The reasons behind this choice were straightforward yet compelling.

GitHub provides an intuitive and user-friendly interface, simplifying the processes of code sharing, reviewing, and merging. This streamlined the development workflow, ensuring that team members could collaborate seamlessly. Its version control capabilities allowed changes to be tracked, conflicts to be managed, and a comprehensive history of the codebase to be maintained, enhancing transparency and accountability.

Moreover, GitHub's robust issue tracking system enabled effective task management and prioritization, particularly valuable in a project of this scale, where numerous features and components needed to be developed and integrated.

2.7.1.4 Cloud Deployment: Amazon Web Services (AWS)

In the modern era of software development, cloud computing has emerged as a game-changer, offering unparalleled scalability, reliability, and flexibility. The benefits of deploying development and production environments in the cloud were recognized, and after careful evaluation, Amazon Web Services (AWS) was selected as the cloud provider.

AWS's vast array of services and global infrastructure ensured that the decentralized social media protocol could scale seamlessly to accommodate increasing user loads. The pay-as-you-go pricing model allowed costs to be optimized while benefiting from world-class infrastructure and support.

Furthermore, AWS offered a range of tools and services for tasks such as server provisioning, load balancing, and auto-scaling. These features simplified the management of cloud infrastructure, freeing up valuable time and resources that could be redirected toward enhancing the protocol's functionality and performance.

2.7.1.4.1 Deployment Using Amazon EC2 Instances

For deploying Docker containers and the PostgreSQL database, an Amazon Elastic Compute Cloud (Amazon EC2) instance was utilized. Amazon EC2 instances offered several advantages, including scalability, flexibility, and ease of configuration. Below are the specifications of the Amazon EC2 instance used for this deployment:

Type	t2.micro
Number of vCPUs	1
RAM (GiB)	1.0
Storage (GiB)	30
Operating System	Ubuntu

Table 2.7.1.4.1.1: Specifications of EC2 instance

2.7.1.5 Containerization: Docker

To achieve consistency in deployment and streamline the process, Docker, a containerization technology, was leveraged. Docker provided several advantages that significantly facilitated the implementation.

Docker's fundamental benefit lies in its ability to encapsulate applications and their dependencies within lightweight containers. Each component of the decentralized social media protocol was integrated and packaged as a Docker image. This approach not only simplified deployment but also ensured that each component could run consistently across various environments.

One of Docker's key strengths is its isolation capabilities, allowing conflicts between different components of the system to be avoided. This isolation reduced the risk of compatibility issues and contributed to the overall stability of the protocol.

Additionally, Docker's portability made it possible to develop and test components independently before seamlessly integrating them into the larger system. This modular approach enhanced development agility and minimized disruptions during the implementation phase.

2.7.1.5.1 Deployment as Docker Containers

The decision to employ Docker containers for deployment resulted in the successful deployment of the implemented demo social platform. This approach offered several tangible benefits:

- **Consistency:** Docker containers ensured that the demo platform ran consistently across different environments, eliminating the notorious "it works on my machine" problem.
- **Scalability:** Docker's scalability features allowed adaptation to varying levels of user demand effortlessly. As user activity increased, the application could be scaled horizontally by adding more containers.

- **Resource Efficiency:** Docker containers, being lightweight and sharing the host OS kernel, resulted in minimal overhead. This translated to efficient resource utilization, reducing infrastructure costs.
- **Quick Deployment:** Docker's quick start-up time enabled the rapid deployment of new updates and features, minimizing downtime and user disruption.

In conclusion, the strategic choices made in setting up the background for implementation were instrumental in ensuring the success of implementation of the decentralized social media protocol. The adoption of Go Lang, PostgreSQL, GitHub, AWS, and Docker laid a strong foundation for the subsequent phases of development and testing. These decisions, rooted in practical considerations and a deep understanding of the requirements unique to the project, enabled the creation of a robust, scalable, and efficient platform that met the goals and expectations.

2.7.2 Setting Up CI/CD Pipeline

Due to the frequent releases during the development process, the team realized the necessity for a CI/CD (Continuous Integration/Continuous Deployment) pipeline. As a result, it was decided to create a reliable CI/CD pipeline that incorporated both CI and CD components. This section describes the pipeline's implementation, which was crucial in accelerating the development and deployment procedures.

2.7.2.1 Continuous Integration (CI) Pipeline

For the CI portion of the pipeline, GitHub Actions was chosen as the preferred tool. This decision was well-founded for several reasons.

GitHub Actions offers a seamless and integrated approach to automating our software development workflows. It allowed us to define, customize, and automate various tasks, such as code compilation, testing, and code quality checks, directly within our GitHub repository.

Moreover, GitHub Actions integrates seamlessly with our codebase hosted on GitHub, making it a natural choice for our CI needs. The ease of configuration and extensive library of pre-built actions simplified the setup of our CI workflow.

2.7.2.2 Continuous Deployment (CD) Pipeline

In parallel with the CI pipeline, a CD pipeline was established using GitHub Actions and Watchtower. Each was chosen for distinct but complementary reasons.

GitHub Actions was extended to serve as our CD tool, ensuring the seamless deployment of our application. GitHub Actions' continuous deployment capabilities allowed us to automate the deployment process, ensuring that every successful CI build was automatically deployed to our production environment. This reduced manual intervention and minimized deployment errors.

Watchtower, on the other hand, played a crucial role in automating container updates. It continuously monitored our Docker containers for new image versions and

automatically updated running containers to the latest versions when changes were detected. This ensured that our application was always running with the most up-to-date code, enhancing security and reliability.

Direct integration to the GitHub repository and the user-friendly configuration of GitHub Actions made them the best option for both CI and CD. It eliminated the need for third-party tools, simplifying our development workflow and ensuring that code changes were rapidly and reliably integrated and deployed.

Watchtower's selection was driven by its ability to automate container updates, reducing the need for manual intervention, and ensuring our application's consistent and secure operation.

The implementation of this CI/CD pipeline provided several benefits including:

- **Automation:** The CI/CD pipeline automated critical aspects of the development and deployment processes, reducing the manual effort required.
- **Speed and Efficiency:** Rapid and automated testing, integration, and deployment improved the overall speed and efficiency of the development cycle.
- **Consistency:** With CI/CD, every code change was subjected to the same automated testing and deployment process, ensuring consistency and reliability.
- **Reduced Errors:** Automation minimized the potential for human errors during deployment, enhancing the overall quality of our application.
- **Frequent Releases:** The CI/CD pipeline facilitated frequent and reliable releases, crucial for the development requiring rapid iterations and updates.

The establishment of the CI/CD pipeline using GitHub Actions and Watchtower significantly enhanced the development and deployment processes, aligning with the unique demands of the project. This automation not only improved efficiency but also

contributed to the consistency and quality of our work, enabling to meet project milestones and deliver robust outcomes.

2.7.3 Development

In the context of data integrity within the decentralized social media protocol, the development phase focused on implementing key functionalities to enhance security and reliability of user data stored on their devices.

Data Integrity through Signature Generation and Verification

To ensure data integrity, the system implemented a process where user data is signed before storage and verified upon retrieval. This involves generating a unique digital signature for each piece of user data. When data is ready for storage, it undergoes a signing process, which creates a unique signature based on the data content and the user's private key. This signature acts as a seal of authenticity and ensures that any tampering with the data can be detected.

Upon retrieval, the stored data is accompanied by its digital signature. To verify the data's integrity, the system checks the signature against the data content and the user's public key. If the signature matches the data, it confirms that the data has not been altered since it was signed, maintaining its integrity.

Data Security through Hashing

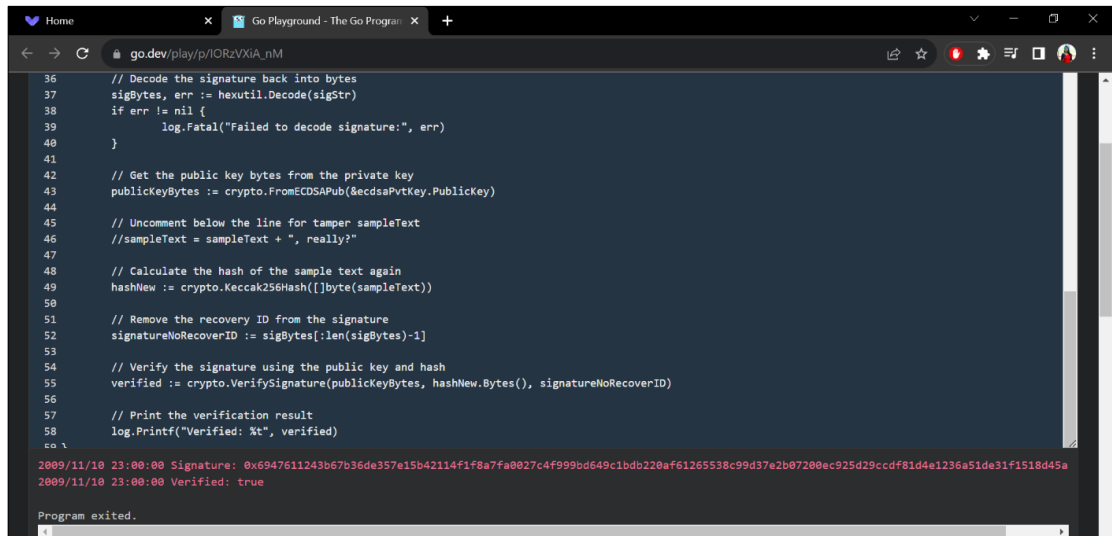
Another critical aspect of data integrity is data security. To enhance the security of user data, a hashing process is employed. When data is ready for storage, it undergoes a hashing procedure that converts it into a unique string of characters, often referred to as a hash or digital fingerprint. This hash is unique to the data's content, and even a small change in the data will result in a significantly different hash. This property ensures that the stored data remains unaltered and secure.

Consistency in Event Representation

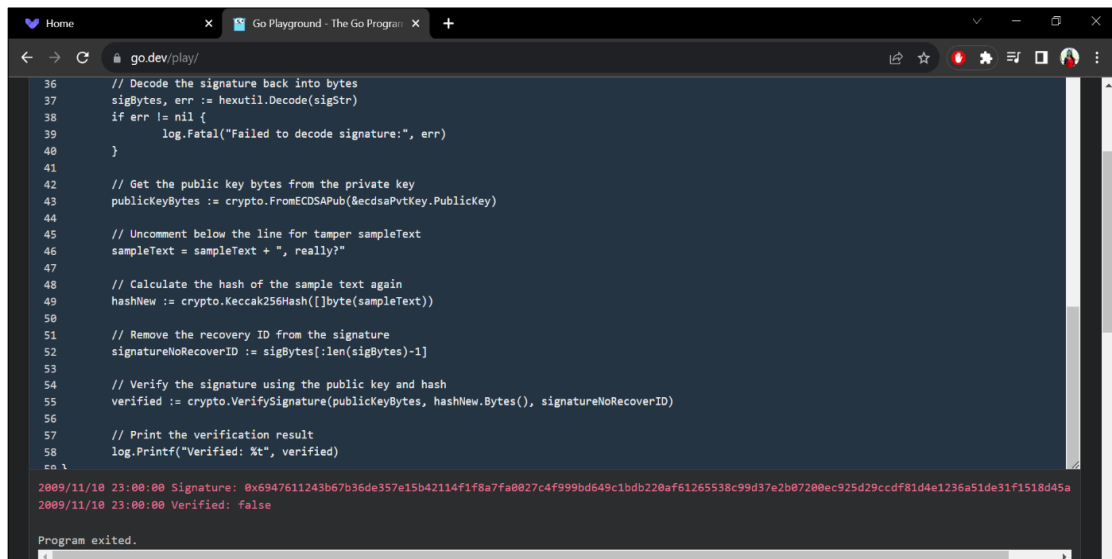
To maintain data consistency and integrity across various user devices, a standardization process is implemented for representing events. Events generated by users are converted into a consistent format before storage and retrieval. This standardized format ensures that regardless of the device used, events are represented

uniformly. By doing so, the system maintains data consistency and enhances the overall integrity of the decentralized social media protocol.

In summary, the development phase for ensuring data integrity in the decentralized social media protocol involved the implementation of processes for data signing and verification, data hashing for security, and standardization of event representation. These processes collectively contribute to safeguarding user data, detecting unauthorized alterations, and maintaining data integrity within the protocol.



```
36 // Decode the signature back into bytes
37 sigBytes, err := hexutil.Decode(sigStr)
38 if err != nil {
39     log.Fatal("Failed to decode signature:", err)
40 }
41
42 // Get the public key bytes from the private key
43 publicKeyBytes := crypto.FromECDSAPub(&ecdsaPvtKey.PublicKey)
44
45 // Uncomment below the line for tamper sampleText
46 //sampleText = sampleText + ", really?"
47
48 // Calculate the hash of the sample text again
49 hashNew := crypto.Keccak256Hash([]byte(sampleText))
50
51 // Remove the recovery ID from the signature
52 signatureNoRecoverID := sigBytes[:len(sigBytes)-1]
53
54 // Verify the signature using the public key and hash
55 verified := crypto.VerifySignature(publicKeyBytes, hashNew.Bytes(), signatureNoRecoverID)
56
57 // Print the verification result
58 log.Printf("Verified: %t", verified)
59
2009/11/10 23:00:00 Signature: 0x6947611243b67b36de357e15b42114f1f8a7fa0027c4f999bd649c1bdb220af61265538c99d37e2b07280ec925d29ccdf81d4e1236a51de31f1518d45a
2009/11/10 23:00:00 Verified: true
Program exited.
```



```
36 // Decode the signature back into bytes
37 sigBytes, err := hexutil.Decode(sigStr)
38 if err != nil {
39     log.Fatal("Failed to decode signature:", err)
40 }
41
42 // Get the public key bytes from the private key
43 publicKeyBytes := crypto.FromECDSAPub(&ecdsaPvtKey.PublicKey)
44
45 // Uncomment below the line for tamper sampleText
46 sampleText = sampleText + ", really?"
47
48 // Calculate the hash of the sample text again
49 hashNew := crypto.Keccak256Hash([]byte(sampleText))
50
51 // Remove the recovery ID from the signature
52 signatureNoRecoverID := sigBytes[:len(sigBytes)-1]
53
54 // Verify the signature using the public key and hash
55 verified := crypto.VerifySignature(publicKeyBytes, hashNew.Bytes(), signatureNoRecoverID)
56
57 // Print the verification result
58 log.Printf("Verified: %t", verified)
59
2009/11/10 23:00:00 Signature: 0x6947611243b67b36de357e15b42114f1f8a7fa0027c4f999bd649c1bdb220af61265538c99d37e2b07280ec925d29ccdf81d4e1236a51de31f1518d45a
2009/11/10 23:00:00 Verified: false
Program exited.
```

Figure 2.7.3.1: Code snippet of data integrity verification

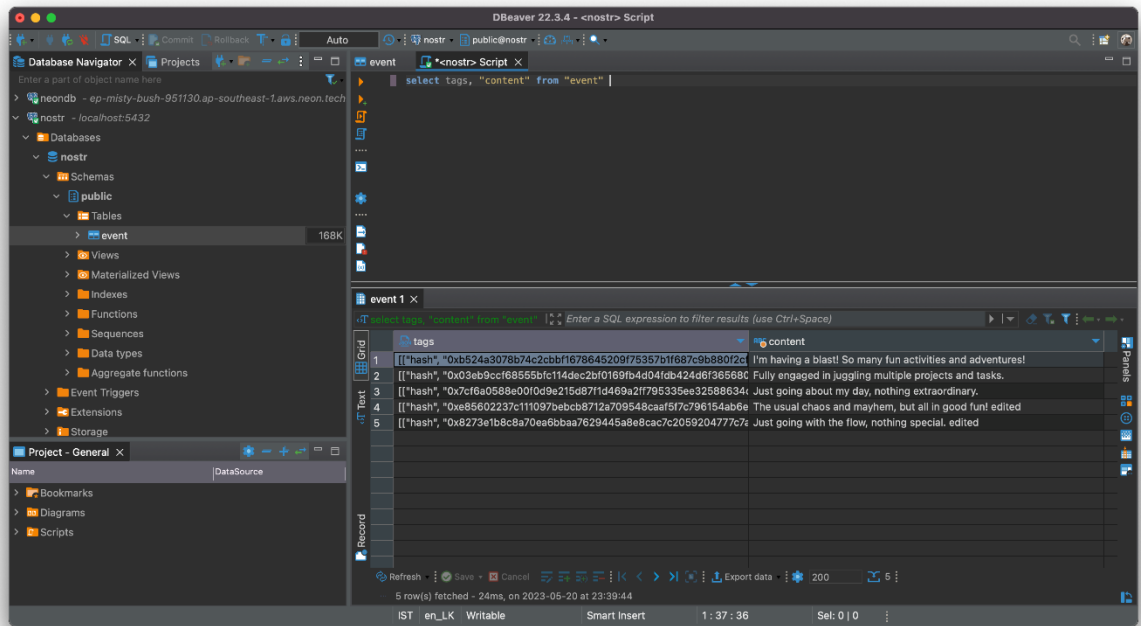


Figure 2.7.3.2: Stored signatures on users database

2.8 Testing

In terms of testing, the approach used was intended to guarantee the accuracy and dependability of the research project. Each development effort was verified through a methodical testing process before it was integrated into the project's codebase hosted on GitHub and after the deployment.

2.8.1 Unit Testing

Unit tests played an important role in the testing process. These tests were conducted before the code was pushed to GitHub, serving as a initial validation step. Unit tests are small, focused tests that verify the correctness of individual components or functions within the code.

Through the conduct of unit tests, potential issues were detected and rectified at an early stage, preventing them from propagating into the codebase. This approach ensured that each development effort underwent validation for correctness and functionality, contributing to the overall stability of the project.

2.8.2 Continuous Integration (CI) and Deployment (CD)

The testing process was tightly integrated with the CI/CD pipeline, streamlining the testing and deployment of developed features. Here's how it worked:

Continuous Integration (CI): Upon completion of development efforts, the CI pipeline automatically integrated these features. It compiled the code, ran unit tests, and ensured that the new code did not introduce regressions or errors.

Docker Image Building: Following successful CI, the pipeline proceeded to build Docker images. Docker images are like packaged containers that encapsulate the application and its dependencies.

Continuous Deployment (CD): A crucial aspect of the testing process was the CD pipeline. This pipeline utilized Watchtower, an automated container updating tool.

When new Docker images were created, Watchtower was triggered as part of the CD process.

Watchtower Deployment: Watchtower's role was to pull the latest Docker images with updated code and deploy them on Amazon Elastic Compute Cloud (Amazon EC2) instances. This automated process ensured that the application consistently ran the latest code changes.

2.8.3 Smoke Testing

After each deployment, a critical step was the execution of smoke tests. Smoke tests are a set of initial tests designed to verify that the newly deployed release is stable and functional. They are meant to ensure that the release is "smoke-free," indicating that it is ready for further testing and use.

The testing approach offered several notable benefits to the project:

Early Issue Detection: Unit tests allowed for the early detection and resolution of issues in the development process, reducing the likelihood of critical errors in the final product.

Quality Assurance: Through thorough unit tests and continuous integration, a high level of code quality and reliability was maintained.

Efficiency: Automation within the CI/CD pipeline reduced manual testing efforts, enabling faster development cycles and quicker deployment of new features.

Consistency: The automated deployment process with Watchtower ensured that the application consistently operated with the latest code changes, enhancing overall reliability.

Rapid Updates: The testing and CD approach facilitated swift updates and deployments, essential for an academic research project requiring frequent iterations and enhancements.

In conclusion, the testing approach, included unit testing, seamless integration with the CI/CD pipeline, and smoke testing, was important in maintaining the quality, reliability, and efficiency of the project. It enabled the validation of each development effort, early error identification, and the delivery of a robust and continuously evolving research platform.

2.8.4 Validation using practical implementations.

In order to assess the practical implementations of the project, the development of a demo social media platform and a separate benchmarking application was initiated. These efforts allowed valuable insights into the performance and functionality of the built protocol.

2.8.4.1 Benchmarking with Infrastructure as Code (IaaC) Approach

To deploy the benchmarking application, along with the required infrastructure and dependent applications, an Infrastructure as Code (IaaC) approach was followed. This approach brought significant advantages:

Reproducibility: IaaC allowed the definition and recreation of the entire infrastructure consistently, minimizing discrepancies between deployments.

Version Control: Infrastructure configurations were version-controlled, enabling the tracking of changes, effective collaboration, and maintenance of a comprehensive history.

Scalability: With IaaC, easy scaling of the infrastructure up or down in response to varying workloads was possible, ensuring optimal performance.

2.8.4.1.1 Terraform as the IaaC Tool

For implementing the IaaC approach, terraform was selected as the tool of choice. Terraform provided numerous benefits, including:

Infrastructure ambiguity: Terraform supports multiple cloud providers and infrastructure types, giving flexibility in choosing the best-suited resources.

Declarative Syntax: Terraform' s declarative syntax made it easy to define and manage infrastructure configurations, enhancing readability and maintainability.

Modularity: Terraform allowed the creation of reusable modules, simplifying the deployment of complex infrastructure components.

Along with these 2 practical implementations, testing is carried out to validate the functionality, usability, and reliability of the protocol.

3 Results and Discussions

3.1 Results

The implementation of both the demo social media platform and the benchmark application played an important role in validating various aspects of the project. Here, present the results obtained from these implementations, highlighting the project's key achievements and insights.

3.1.1 Demo Social Media Platform Results

The development of the demo social media platform on top of the implemented decentralized social media protocol produced below noticeable results:

User Engagement: The platform successfully facilitated user engagement, demonstrating the protocol's effectiveness in creating a user-friendly social media environment.

Feature Integration: Various features, such as user profiles, posts, and interactions, were seamlessly integrated and validated, showcasing the versatility of the protocol.

User Experience: Users were provided a positive and intuitive experience while navigating and interacting with the demo social media platform, highlighting its user-centric design.

Stability: The platform established stability and robustness, with minimal downtime or disruptions during usage, ensuring a reliable user experience.

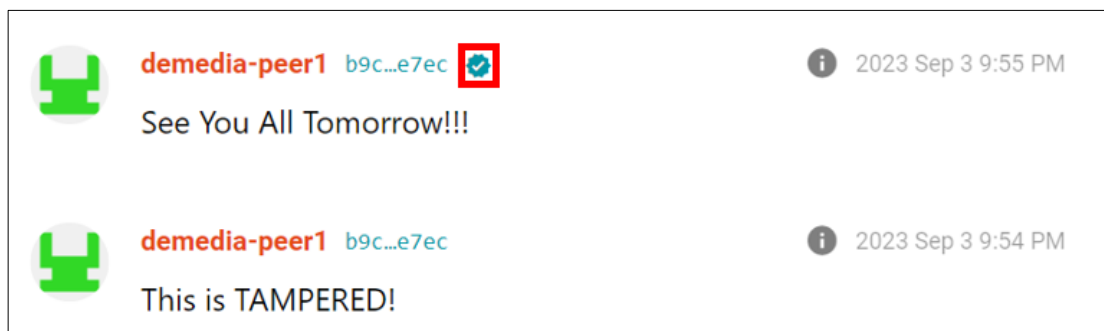
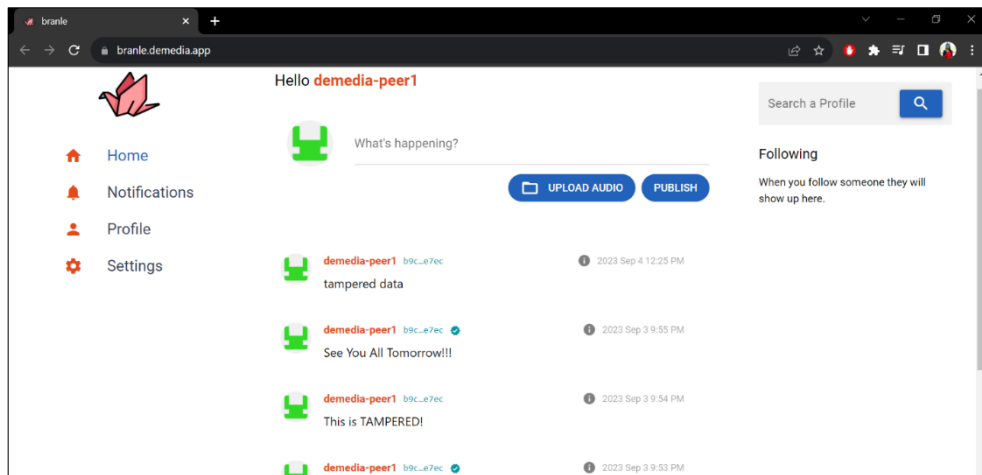


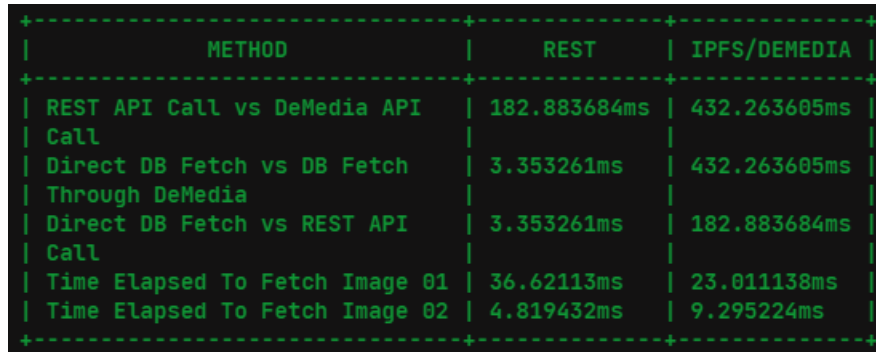
Figure 3.1.2.1: Screen capture of data integrity on demo social media platform

DeMedia Panel			
Hubs	Decentralized Data	Data Integrity	Decentralized Caching
ID	CONTENT	HASH	IS VERIFIED
e2e56e2a33b6edee0afea07f15f382d73d3648b8af3dbddf24d1060cd25bb1a6	tampered data	0x42d924931b3d276b9249daf5d92fd602a0c15044973e824b3b5de0d3efe6af6d6d2e1408624cf0efd4c51633d623a45e0b0cd6a83a0f92b05ae245dacca80bb801	false
de5bd1f237e49dee13e427498127a1f33b3660c53f073998147d83c2836f6632c	See You All Tomorrow!!!	0x1497ef61fe7e6832c3df31073fd0d39905a88ccea282840cbbc9bf0cada71d21251ea72b21c45f4ee098d515cflb7dd017e48aecbbce7fe6828c18055722d9ee01	true
812d7ab03147910220a71f23908636cfe5d2b31ba4397c4488a44d4c7f20f2c9	This is TAMPERED!	0x634cd76562d7404b24080abc1e12952ea2d90988e542597e5931e42ef4b9d0bc13134d3d3d8301d731ae1f8ec83a505aa352a9d6cc1ba8a2e55437e0b5d4431e01	false

Figure 3.1.2.2: The panel of Decentralized Data integrity

3.1.2 Benchmarking Results

The benchmarking application provided insightful results on the performance and capabilities of the built protocol. These results, which are detailed below, are integral to the project's evaluation and future improvements:

A terminal window with a black background and green text. It displays a table of benchmark results. The table has three columns: METHOD, REST, and IPFS/DEMEDIA. The data is as follows:

METHOD	REST	IPFS/DEMEDIA
REST API Call vs DeMedia API Call	182.883684ms	432.263605ms
Direct DB Fetch vs DB Fetch Through DeMedia	3.353261ms	432.263605ms
Direct DB Fetch vs REST API Call	3.353261ms	182.883684ms
Time Elapsed To Fetch Image 01	36.62113ms	23.011138ms
Time Elapsed To Fetch Image 02	4.819432ms	9.295224ms

Figure 3.1.3.1: Output from benchmark application

3.1.3 Results from overall practical implementations

The development of the demo social media platform and the benchmark application allowed for the validation of several critical aspects of the project, including:

Functionality: Through rigorous testing and usage of the demo social media platform, validated the core functionality of the implemented decentralized social media protocol.

Scalability: By deploying the demo platform and benchmark application at scale, assessed the protocol's ability to handle a substantial user load and interactions effectively.

Security: Security features and measures were thoroughly examined and validated through the development of the demo platform, ensuring the protection of user data and interactions.

Performance: Performance metrics were gathered and analyzed to evaluate the speed and efficiency of the decentralized social media protocol in real-world scenarios.

3.2 Research Findings

Several key findings came out from my exploration of the data integrity component within a decentralized social media protocol. The decentralized structure easily recognizes unauthorized data modifications, and tampered data. It increases trust among users who utilize the protocol.

This decentralization also minimized single points of failure by distributing data across various nodes and reducing the overall risk of data breaches. And the use of cryptographic signatures provided users with an effective way for verifying the authenticity and integrity of their data. As the network expands, the protocol's ability to maintain data integrity without compromising performance would be a crucial finding.

Understanding how the protocol adapts to the expanding requirement for a network while maintaining data integrity and ensuring smooth performance will be essential to its long-term success.

3.3 Discussion

The protocol's commitment to keeping data safe and reliable was demonstrated through a verification mechanism built into the demo platform. Hubs play an important role in maintaining the integrity of data by thoroughly checking the cryptographic signatures associated with users' content. These cryptographic signatures behave as digital fingerprints, generated uniquely when content is initially stored on a user's device and then transferred over the network. Hubs authenticate the validity and unchanged nature of the data through this verification process, providing users confidence that the information they gave is trustworthy and has not been tampered with during transmission.

The verification process had a significant impact on the platform's users by offering them a guarantee regarding the authenticity and unaltered state of the shared data. In practical terms, this meant that individuals using the platform could have full confidence that the information they accessed or interacted with was legitimate and had not been tampered with in any manner. It provided users with a sense of trust. This ensures a more secure and reliable user experience within the platform and enhances its overall popularity and importance.

4. CONTRIBUTION SUMMARY

IT number	Name	Tasks
IT20137496	Dhananjani G.G.S.	<ul style="list-style-type: none">● Conducted a comprehensive review of data integrity preservation mechanism and associated research.● Developed a cryptographic mechanism to save data in user's device.● Generated hash value for user data using the hashing algorithm.● Signed hash values using digital signature and encoded it.● Maintain data integrity of stored set of signed values in IPFS network.

Table 3.0.1: Personal and Facilities

5. CONCLUSION

The main objective of the proposed research component is to provide a data integrity mechanism to preserve user data in decentralized social media protocol. Data integrity is a critical aspect of any decentralized social media protocol, and DeMedia provides a robust mechanism to ensure the integrity of user data. It provides users security, privacy, and a trustworthy environment for users to share and communicate without having to worry about the security of their data.

By utilizing cryptographic hashing and data encoding, DeMedia establishes unique hash values for each piece of user-generated content or data, making it difficult for data to be tampered with or corrupted by external parties. This helps to build trust and confidence among users, as they have control over their data and can verify its legitimacy. DeMedia also offers both quantitative and qualitative measures to evaluate data integrity, allowing for a comprehensive assessment of the accuracy, consistency, and reliability of the data stored on the IPFS network.

DeMedia's data integrity mechanisms provide several advantages. First, they allow even non-technical users to easily deploy peers on their devices with low operational costs, making it accessible for a wide range of users. Second, by giving users full control over their data and enabling them to make changes at their discretion, DeMedia reduces the risk of data corruption or tampering by outside parties. This empowers users to have greater ownership and control over their data, enhancing their trust in the platform.

As a summary, DeMedia's data integrity mechanisms provide a secure and trustworthy environment for users to share and communicate on a decentralized social media platform. By ensuring the integrity of user data, DeMedia enhances user confidence in the platform, mitigates the risk of data tampering or corruption.

REFERENCES

- [1] <https://brand-experience.ieee.org/guidelines/social-media/social-media-overview/#:~:text=IEEE%20social%20media%20consists%20of,and%20respond%20to%20th at%20content, IEEE>
- [2] Bogdan Tiganoaia, Alexandra Cernian, Andrei Niculescu, "The use of social platforms and personal data protection - An exploratory study" IEEE, 2017.
- [3] Tharuka Sarathchandra, Damith Jayawikrama, "A decentralized social network architecture", IEEE 2021.
- [4] Maryam Qamar, Mehwish Malik, Saadia Batool, Sidra Mehmood, "Centralized to Decentralized Social Networks", ResearchGate 2016.
- [5] Shuai Zeng, Yong Yuan, Fei-Yue Wang, "A decentralized social networking architecture enhanced by blockchain", IEEE 2019.
- [6] Biwen Chen, Libing Wu, Huaqun Wang, Lu Zhou, Debiao He, "A Blockchain-Based Searchable Public-Key Encryption With Forward and Backward Privacy for Cloud-Assisted Vehicular Social Networks", IEEE 2019.
- [7] Kelsie Nabben, "Decentralized Technology in Practice: Social and technical resilience in IPFS", IEEE 2022.
- [8] Subhashish Mazumdar, "From data integrity to inference integrity", IEEE 2017.
- [9] Vishwanath G. Garagad, Nalini C. Iyer, Heera G. Wali, "Data Integrity: A security threat for Internet of Things and Cyber-Physical Systems", IEEE 2020.
- [10] Ting Cai, Zicong Hong, Shuo Liu, Wuhui Chen, Zibin Zheng, Yang Yu, "SocialChain: Decoupling Social Data and Applications to Return Your Data Ownership", IEEE 2017.
- [11] Chhavi Tuteja, Nitesh Saxena, Prashant Johri, Vikas Rao Vadi, "Blockchain Technology: A case study of its Decentralized Use", IEEE 2022.
- [12] Ruibo Cao, Kun Meng, Kai Sun, Ziqiang Zheng "Evaluation Model of Data Consistency Mechanism in Decentralized Network Application", IEEE 2021.
- [13] "Data Integrity-Based Methodology and Checklist for Identifying Implementation Risks of Physiological Sensing in Mobile Health Projects: Quantitative and Qualitative Analysis", No.12, 2018.

- [14] Reza Nourmohammadi, Kaiwen Zhang, "An On-Chain Governance Model Based on Particle Swarm Optimization for Reducing Blockchain Forks", IEEE 2022.
- [15] Thomas Paul, Antonino Famulari, Thorsten Strufe, "A survey on decentralized Online Social Networks", ScienceDirect, 2014.
- [16] [Online] available: <https://ethtps.info>
- [17] Quanqing Xu, Zhiwen Song, Rick Siow Mong Goh, Yongjun Li, "Building an Ethereum and IPFS-Based Decentralized Social Network System", IEEE 2018.
- [18] Van-Duy Pham, Canh-Tuan Tran, Thang Nguyen, Tien-Thao Nguyen, Ba-Lam Do, "B-Box - A Decentralized Storage System Using IPFS, Attributed-based Encryption, and Blockchain", IEEE 2020.
- [19] Manjula K Pawar, Prakashgoud Patil, Manisha Sharma, Megha Chalageri, "Secure and Scalable Decentralized Supply Chain Management Using Ethereum and IPFS Platform", IEEE 2021.
- [20] [Online] available: <https://www.deso.com/>
- [21] Julija Golosova, Andrejs Romanovs, "The Advantages and Disadvantages of the Blockchain Technology", IEEE 2018.
- [22] Nursena Baygin, Mehmet Baygin, Mehmet Karakose, "Blockchain Technology: Applications, Benefits and Challenges", IEEE 2019.
- [23] R. Keefe, "5 popular project management methodologies and when to use them," Toggl Blog, 13-Sep-2022. [Online]. Available: <https://toggl.com/blog/5-popular-project-management-methodologies-and-when-to-use-them>. [Accessed: 20-Mar-2023].

GLOSSARY

- Centralization - A system in which power and decision-making are concentrated in a single entity or a group.
- Decentralization - A system in which power and decision-making are distributed among multiple entities or individuals.
- Integrity - Ensuring that it is accurate and has not been changed.
- Encrypt - To convert information or data into a secret code to prevent unauthorized access or interception.
- Decrypt - To convert an encrypted code back into its original, readable form.
- Cryptography - The practice of secure communication using codes and ciphers.
- Peer - A computer or device connected to a network that shares data and resources with other peers.
- Hub - A central point or device that connects other devices or things together.
- Caching - The process of storing frequently accessed data in a temporary storage location to improve access time and reduce network traffic.
- Network - A group of interconnected devices and communication channels that enable communication and data exchange.
- Protocol - A set of rules and procedures that govern the communication and exchange of data between devices on a network.
- Algorithm - A step-by-step set of instructions for solving a problem or performing a task.
- Hash - A unique numerical value generated by a mathematical function that represents a digital file or data.
- Signature - A special code or stamp that proves a document or message is real and has not been changed by someone else.
- Blockchain - A decentralized, distributed digital ledger that records transactions across multiple devices and networks.
- Public key - A unique cryptographic code that is available to everyone and used to encrypt data.

- Private key - A unique cryptographic code that is kept secret and used to decrypt data.
- Cipher - An algorithm used to encrypt or decrypt data.
- Hacker - A person who gains unauthorized access to computer systems or networks with malicious intent or for personal gain.

APPENDICES

Data Integrity			
ORIGINALITY REPORT			
3 %	1 %	1 %	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Sasha Shilina. "chapter 10 The Promise of Blockchain-Based Decentralized Social Networks", IGI Global, 2023 Publication	1 %	
2	Submitted to TAFE Queensland Brisbane Student Paper	<1 %	
3	essay.utwente.nl Internet Source	<1 %	
4	Dipti Trivedi, Venkataramana Badarla, Ravi Bhandari. "Occupancy inference using infrastructure elements in indoor environment: a multi-sensor data fusion", CCF Transactions on Pervasive Computing and Interaction, 2023 Publication	<1 %	
5	Submitted to University of Ruhuna Matara Student Paper	<1 %	
6	uwspace.uwaterloo.ca Internet Source	<1 %	
7	Submitted to NCC Education		