DEMEDIA – DECENTRALIZED SOCIAL MEDIA PROTOCOL

Abeykoon A. W. Y. I. K.

IT120157432

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

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

MECHANISM FOR DATA CACHING PRESERVATION

Abeykoon A. W. Y. I. K.

IT120157432

The dissertation was submitted in partial fulfillment of the requirements for the B.Sc. Special Honors degree in Information Technology Specialized in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

DECLARATION

I declare that this is my own work, and this dissertation 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 my 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 non-exclusive 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)

Name	Student ID	Signature
Abeykoon A.W.Y.I.K.	IT20157432	

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

………………………….. ………………………….

Signature of the supervisor: Date

(Mr. Kavinga Yapa Abeywardena)

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.

ABSTRACT

A decentralized social media protocol is a collection of rules and standards for creating social media networks without the need for a central authority or intermediary. These protocols enable users to engage with one another without depending on a centralized service provider by using decentralized technologies such as Blockchain, peer-to-peer networking, and distributed storage. “DeMedia” is a decentralized social media protocol that aims at giving users complete control over their data and utilizes peer-to-peer (P2P) networking rather than a blockchain to cut expenses. The protocol allows users to control their data by using a client application that keeps the data locally on their device rather than on a central server. Users have full control over the length of caching, which is utilized to guarantee optimal speed. Moreover, the protocol uses cryptographic mechanisms to ensure data integrity. “DeMedia” provides a complete decentralized social media solution that prioritizes user data ownership and control above a costly blockchain infrastructure. This paper describes a decentralized social media platform protocol aimed at addressing centralized social media platform problems such as data privacy, censorship, and data ownership. The protocol is built on peer-to-peer communication, which allows users to connect without the use of intermediaries. The platform contains a user data decentralization protocol, which allows individuals to maintain ownership over their data while assuring that no centralized authority owns or controls it. This platform also has a data integrity preservation mechanism to ensure the integrity of user data. Additionally, the platform has data cache preservation technology which aids platform speed by keeping frequently visited information locally on users' devices. Finally, the suggested platform protocol provides a decentralized and secure social media network in which users control their data and connect without the necessity of centralized intermediaries.

Keywords: protocol, data cache, decentralization, peer-to-peer

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

IPFS	Interplanetary File System
AWS S2	Amazon Web Services
CDN	content delivery network
MFS	Mutable File System
CID	Content Identifier

1.0 INTRODUCTION

Social media refers to a wide range of digital technologies that allow people to share ideas and information through virtual networks and communities. With 4.7 billion users globally, social media has become an essential part of modern life, connecting people, and building communities. It has, however, been criticized for disseminating false information and violent speech. Regardless, it has evolved into an important component of companies' marketing efforts, helping them to reach and engage their target audiences. Leading global platforms that shape communication and interaction include Facebook, YouTube, WhatsApp, Instagram, and WeChat. [1] Because of changing industries and technological advancements, social media is going through incredible growth. Platforms such as Facebook, Twitter, and TikTok are moving toward premium subscription-based services that provide customized content, a safer online environment, greater privacy safety features, and building communities. Video marketing, a $135 billion industry by 2020, is quickly becoming the dominating channel for content. Businesses must align with these trends while retaining their distinctive brand identity. The underlying drive for human connection and engagement will influence social media's future. [2]

Centralized social networking platforms have limitations such as creating information silos and lack of user control over personal data. They restrict data interoperability and have a history of misusing user information. A decentralized approach to social networking is proposed, built on open technologies like Linked Data, Semantic Web ontologies, and open single-sign-on systems. This framework uses Uniform Resource Identifiers (URIs) to create a distributed, extensible system that offers users increased control over their data. The motivation for this approach comes from the proprietary nature of existing sites, which control user data for targeted advertising and financial benefits. Decentralized networks provide greater user autonomy and offer opportunities for creative application development. They can also serve as an infrastructure for social resource discovery, connecting users across platforms based on shared interests. [3]

Figure 1.1: Web 1.0 -> Web 2.0 -> Web 3.0?

The evolution of the internet has shifted from Web 1.0's static "read-only" pages to Web 2.0's dynamic "read-write" platforms marked by user-generated content and social media engagement. Web 2.0, however, also brought with it difficulties like centralization, privacy concerns, and the commercialization of user data. As the Web3 age begins, the focus has shifted to a "Read-Write-Own" model that aims to give individual users more power and control. Web3 offers decentralized applications (dapps), Decentralized Autonomous Organizations (DAOs), and self-governing identity methods based on Blockchain technology, all supported by crypto-currency systems. Despite challenges such as high transaction fees and complexity, Web3's promise resides in genuine digital asset ownership, Adaptability to censorship, and inclusive decentralized governance, providing a path to a more transparent and user-focused digital world. [4] [5]

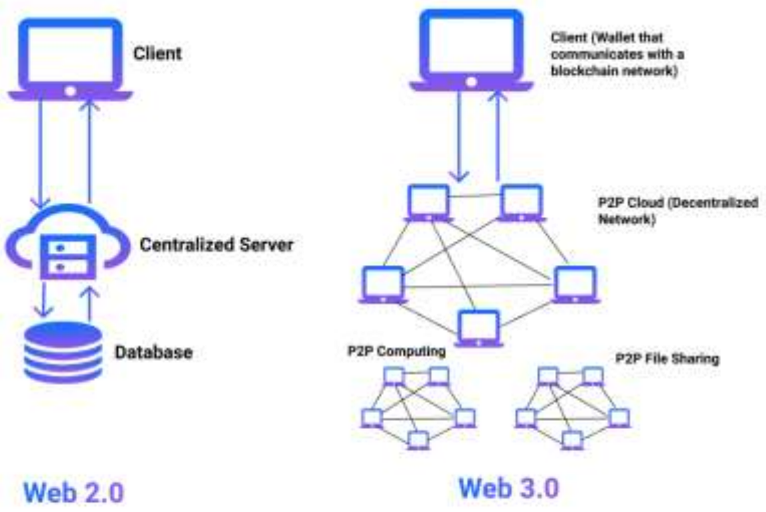

Figure 1.2: Difference of the web 2.0 and 3.0

DeMedia is a decentralized social media protocol that allows social media platforms to control a decentralized network, removing the need for a centralized server controlled by a single platform. [6]

One of the most important aspects of a decentralized social media network is its caching mechanism. Data caching is the process of keeping several copies of data or files in a temporary storage space—also known as a cache—so they can be viewed more quickly. In order to guarantee users do not need to download information each time they visit a website or application to speed up site loading, it saves data for software applications, servers, and web browsers. Because there is no central server to store the data, it is crucial to implement a caching method in a decentralized social media platform. Data that is commonly accessed is cached and kept close to the user.

1.1 Background Literature

Thousands of people use social media on every day of their lives. They could not survive without it because most duties in the current world are time-based and take place on these social networks. These networks typically concentrated. It allows socially conscious businesses to monitor and evaluate data. It could be a reason for questioning the security of the users' data. Furthermore, these companies frequently distribute them to third parties while making substantial profits without the customer's authorization. Many people have raised this issue because data is the most valuable resource in both the present and the future. As a result, attention has been turned to decentralized social networking platforms where people can save their data locally. [7]. The network resources and services are distributed through the network called "distributed" and it might be geographically distributed over the network. The network and resources are managed by a central authority, and they have the power to do everything in the system. Decentralized refers to a situation in which there is no central location or supervisor with authority over the system. Users of the system are controlled by the system.

Traditional Web2 social media platforms, centralized by design, grapple with challenges such as censorship, privacy breaches, and demonetization. This has facilitated the change to the decentralized Web3 model, which provides a more representative digital world. For example, Reddit is at the top of this modification, introducing ERC-20 tokens for content contributions and using the Arbitrum protocol for increased scalability. Similarly, giants like Twitter and Meta are exploring the potential of Non-Fungible Tokens (NFTs) to augment user engagement. This paradigm shift, as detailed by Cointelegraph is rooted in Blockchain technology, emphasizing user privacy, data ownership, and resistance to censorship.

Decentralized platforms, such as Diaspora, Mastodon, and Hive, are creating this space, distributing user data across multiple nodes for heightened security and transparency. While challenges such as scalability, user adoption, and legal complications are around, the general direction toward Web3 demonstrates a commitment to security, confidentiality, and the freedom of communication. As the digital ecosystem evolves, it's evident that the demand for platforms prioritizing user-centricity will surge, reshaping the future of online interactions and content dissemination. [8]

1.1.1 Examples for decentralized social media platforms

Social Media Platform	Brief Description	Storage Network	How cache works
Peepath [9] *Figure 1.1.1.1: Peepat Logo*	It consists of two components: an open-source smart contract that runs on the Ethereum blockchain (data storage) and Peepeth.com. (a front-end).	IPFS	Metadata and optional file caching
Mastodon [10] *Figure 1.1.1.2: Mastodon Logo*	Software for running self-hosted social networking services called Mastodon is open-source and free. Similar to Twitter, it has features for micro blogging.	Federated network of Mastodon instances	No built-in caching mechanism
Hive [11] *Figure 1.1.1.3: Hive Logo*	Hive is a forward-thinking decentralized blockchain and ecosystem that is built to expand with universal adoption of the currency and platforms in mind.	Hive Blockchain	Using LevelDB and RocksDB

Steemit [12] *Figure 1.1.1.4: Steemit Logo*	Steemit is a decentralized application (DApp) built on the Steemit blockchain that rewards users for their content with the eponymous cryptocurrency STEEM. Users can decide the payout of posts and comments by voting on them.	Steemit Blockchain	Using LevelDB [13] and RocksDB [7]

Table 1.1.1.1: Features of decentralized social media

1.1.2 Examples for decentralized social media platforms

Decentralized storage networks need to be discussed when discussing decentralized social media platforms. With decentralized storage, information is encrypted and distributed across several locations, or nodes, that are managed by people or businesses that sell off any extra disk space they have. It is now acting as a challenger to established online storage models. The Decentralized storage networks can have a significant impact on the caching system used by decentralized social media platforms. Many companies presently offer decentralized storage options, and some of them use Blockchain technology to speed up storage operations and validate transactions across a distributed network. Cost savings are one advantage of decentralized storage networks. Since a decentralized network serves as a middleman between those who need storage and those who have extra capacity accessible for leasing, organizations may experience lower costs by primarily using it for storage. Decentralized storage is consequently less costly than conventional cloud storage. Decentralized storage network goods come in a variety. (Arweave, BitTorrent, Filecoin, MaidSafe and Safe Network, Sia, Storj, Utopia) [14]

1.1.2.1 Arweave

Figure 1.1.2.1.1: Arweave Logo

The Arweave protocol makes use of a Blockchain-like structure known as block weave to provide a mechanism for permanent on-chain data storage as well as storage payment. A block in the block weave points to the preceding block and a recall block that is deterministically chosen based on the previous block's information. [15] While the weave is immutable and its data is censorship-resistant, each node has the option to refuse accepting content. Refusing content by a large enough number of nodes prevents unwanted content from being included. [16]

1.1.2.2 BitTorrent

Figure 1.1.2.2.1: BitTorrent Logo

We can define BitTorrent as a peer-to-peer file-sharing protocol. It has an incentive system that manages download behaviors to ensure appropriate resource usage. BitTorrent was designed to provide a more efficient way to distribute rather than a centralized server. This is accomplished by taking advantage of the fact that data are replicated with each download, allowing for self-scalability in file distribution. [16]BitTorrent Fie System designed to reduce costs, improve fault tolerance and avoid government censorship. [14]

1.1.2.3 Storj

Figure 1.1.2.3.1: Storj Logo

Storj aims to be a decentralized cloud storage service. Storj Labs Incorporated intends to compete with centralized storage providers. Storj offers compatibility with the Amazon S3 Application Programming Interface to increase general adoption and ease migration for new users. [17] Because Storj offers Cloud Storage, users can store and retrieve data as well as delete, move, and copy data. Storj is based on Blockchain technology [18]

1.1.2.4 IPFS

Figure 1.1.2.4.1: Arweave Logo

IPFS is a modular set of protocols for organizing and sharing data that was built from the ground up with content addressing and peer-to-peer networking in mind. Because IPFS is open source, there are numerous IPFS implementations. While IPFS has other use cases, its primary use case is for decentralized data publication (files, directories, webpages, etc.). IPFS network nodes can automatically cache resources they download and make them available to other nodes. This system is dependent on nodes being willing and capable of caching and sharing network resources. Because storage is limited, nodes must purge part of their previously cached resources to create a place for new resources. This is known as garbage collection. [19]

IPFS pinning services are essential for data preservation and availability within the IPFS network. Unlike IPFS nodes' default behavior, which stores data in a cache that may be removed, pinning services allow users to identify their data as significant, preventing it from being removed from the network. This capability is especially useful for devices such as phones and tablets, which may only have limited access to the IPFS network. It's also important in private networks, because transfer of information is dependent on unique protocols and network node resources. However, focusing only on network nodes may not be sufficient in situations with high network termination rates or in decentralized networks with limited resources. Pinning services provide an option by frequently using permanent host nodes on cloud service providers to store data securely. However, users must have faith in these systems because they rely on either paid services or freely selfless peers to keep their data. Although pinning services are expensive, they have benefits such as faster data retrieval. This speed improvement results from IPFS's capacity for distributed data retrieval and the strong connectivity of pinning services to the IPFS network. It becomes clear that some providers offer additional functionalities when comparing various pinning services connected with IPFS. Pinata and Temporal, for instance, offer special features like creating private networks for more data privacy and encrypting data for increased security. On the other hand, although it has fewer functions, is easier to use and simpler. Users should be suggested that, regardless of the particular service they use, these services often function on subscription plans with prices dependent on the amount of data pinned. As a result, they are useful resources for anyone looking for dependable data storage and accessibility within the IPFS network. [20]

1.2 Research Gap

This research aims to address the current limitations and problems of decentralized social media systems. It will examine the consequences of storing a decentralized social media protocol using a caching mechanism specifically. Existing decentralized social media protocols can use caching mechanisms to improve performance and reduce network load, but there are some issues with existing protocols such as data consistency, privacy concerns, issues of caching strategies, network congestion, and scalability issues. We proposed IPFS to solve these issues, providing a distributed and decentralized infrastructure for content delivery, potentially reducing reliance on social networks. Users may find it more attractive to participate in content caching and delivery if they do not have to rely on a certain social network.

Steemit and Peepath are two examples of decentralized social media platforms that use blockchain, but it raises challenges such as crime, volatility, and storage issues. Even though they use caching mechanisms, there are still problems such as crime, volatility, and storage issues [21]. DeMedia suggests using IPFS as a storage network, as it has an aggressive caching mechanism that keeps items local for a short period of time. However, these objects may be garbage-collected on a frequent basis, so it is important to pin or add the CID to MFS to prevent garbage collection. [22]

IPFS is a peer-to-peer distributed file system designed to provide a more durable and decentralized system for saving and distributing files on the internet. Caching mechanisms are important for content management, and IPFS caching and peer-to-peer networking techniques can be used to manage the cache. This can help store frequently requested information closer to users, improving speed and decreasing delay. [19]

	Mastodon	Steemit	Peepath	DeMedia
Caching	No	Yes	Yes	Yes
Decentralized Data Storage Network	No	IPFS	Swarm	IPFS
Blockchain based	No	Yes	Yes	No
Scalability	No	Yes	Yes	Yes
Resource Consumption	low	high	Medium	low

Table 1.2.1: Comparison of existing decentralized social media platforms

1.3 Research Problem

Decentralized social media networks need a caching system to improve their usefulness and scalability. In order to speed up response times and lessen network load, the platform may swiftly fetch and display frequently accessed content for users by caching user data. In addition, caching methods can improve the user experience by reducing network load and scalability issues. Decentralized social media networks must use a reliable caching technique to retain optimum performance, scalability, and user experience.

There are so many issues on decentralized social media platforms when we don't use caching mechanisms correctly. The major problems are inconsistent data, cache invalidation, storage capacity, security vulnerabilities, and scalability issues. So, it is very important to identify the most suitable way to implement the caching mechanism on decentralized social media platforms.

As I have identified, the majority of current platforms depend on Blockchain technology while considering the caching system and storage network of decentralized social media platforms. [23] However, there may be several problems with existing decentralized systems implemented on Blockchain. [24] Some disadvantages of implementing Blockchain in decentralized systems include high resource consumption, limited storage capacity, and slow transaction speed, difficulty in cache invalidation, high cost, and lack of adoption, low scalability, and low workforce availability. It impacts user behavior through the caching mechanisms of decentralized social media platforms.

There are several problems with the caching mechanism on previously decentralized social media platforms. We have to implement a better caching mechanism to solve the issues on the existing platforms.

1.4 Research Objectives

1.4.1 Main Objective

The main objective of this research is to provide a reliable solution for storage management concerns in existing decentralized social media platforms by implementing caching mechanisms. This component is important because there are various issues with the caching mechanism in decentralized social media protocols. As a solution to the above problem, we can use IPFS (Interplanetary File System) without using Blockchain based technology. The reason is that the cost of the Blockchain technology, as we are going to deploy it using a free gateway.

Our primary goal in this comprehensive study research is to develop an improved way to resolve the ongoing data management difficulties that affect modern decentralized social media systems. Our research is focused on the strategic deployment of effective caching methods, which have become known as an essential area in resolving the limitations of existing decentralized social media protocols. The complexity of data caching in decentralized environments has contributed to the demand for development. To address this difficulty, we maintain the implementation of IPFS (Interplanetary File System), an advanced distributed file system, without being limited by the limits of Blockchain technology. Our effort has an opportunity to transform the fundamental basis of how data is stored, accessed, and controlled inside decentralized social media networks.

We want to improve decentralized social media by utilizing IPFS and similar technologies properly. We desire it to be faster, and more reliable and put users at the center of it all. This study is expected to alter how we keep information in decentralized social media platforms, and it will have a significant impact on how these platforms function in the future. We're improving storage, which is a major issue.

1.4.1 Sub Objectives

- Get familiar with using cache mechanisms for storage management.

This subtask aims to understand the fundamentals of storage management using caching mechanisms, including how it works, different types of caching mechanisms, and their pros and cons. It also focuses on different use cases and best practices for implementing and managing caching in a storage system.

- Investigate the effects of the decentralized social media protocol's caching mechanism.

This subtask investigates how caching strategies can increase performance, reduce latency, and minimize bandwidth requirements for decentralized social media systems. It is critical to assess the potential drawbacks of implementing caching in a decentralized setting.

- Examining how decentralized social media protocol's caching techniques can be used to manage storage.

This subtask investigates how caching strategies can be used to enhance performance and reduce resource usage in decentralized social media protocols. It includes assessing how caching can be used to manage user-generated content, optimize storage of media assets, and minimize data movement between nodes.

- Integrating the caching mechanism with the IPFS.

This subtask investigates how caching systems can be combined with IPFS to improve efficiency and reduce resource usage in decentralized social media protocols. It involves understanding how to combine caching with IPFS content addressing and retrieval and assessing various caching techniques to maximize IPFS storage.

- Reduce resource consumption in data storage.

This subtask involves developing methods to reduce data storage resources for decentralized social media protocols. It involves assessing the use of caching technologies to decrease resource consumption, optimize data retrieval, and reduce storage needs. It is important to consider how caching can be used with other methods.

- Documenting caching mechanism and integration with IPFS to facilitate adoption and optimization.

The final objective is to document the caching mechanisms and integration with IPFS to facilitate adoption and optimization of decentralized social media protocols. This includes creating documentation that outlines best practices for using caching in a decentralized environment, providing guidance on how to integrate caching with IPFS, and providing examples of successful caching implementations in decentralized social media platforms.

2.0 METHODOLOGY

2.1 Requirement gathering

The first phase in developing a decentralized social media protocol with a caching mechanism is gathering the requirements of the protocol and client apps used to connect to peer-to-peer networks. One of the most difficult challenges is ensuring that cached content is up to current and consistent across all network nodes. To address these issues, protocols and client applications must prioritize scalability, performance, security, and usability. Finally, the client application should have an easy-to-use interface.

The analysis is performed to identify the challenges in present research and the improvements that need to be made. The research criteria are usually gathered by a review of previously published research articles in the topic area. In addition, I searched several blogs, papers, and industry professionals in this subject to learn more about related works. The primary goal of requirement gathering is to define and comprehend the study issue, as well as previously used methodologies, approaches, procedures, and algorithms.

2.1.1 Past Paper Analysis

Analyzing research articles will help with data collection. Analyzing previously published research articles allows for an analysis of current solutions as well as the advantages and disadvantages of past methodologies. Analyzing previous research is useful for selecting the research procedure. Peer-to-peer network, data integrity, caching mechanism and other specialized topics were all examined in the research publications.

2.1.2 Refer Official Documentations

The official documentation is a valuable resource for staying informed about the technology we intend to use in our system development. Unlike previous research publications that may contain outdated information, official documents provide realtime and accurate insight into the latest updates, features, and best practices in technology. It's written and maintained by experts who understand the technology in-house, provides detailed coverage, troubleshooting guides, and essential security updates, making it an essential reference for building an effective and up-to-date system.

2.1.3 Identify Existing Methodologies

This is especially useful when analyzing present systems because it allows for identifying research requirements. It also helps in the analysis of the limitations and weaknesses of existing systems and their application to the newly developed system. Existing systems can be found online, and our research will help to design a high-quality solution that is free of current weaknesses and limits.

2.2 Feasibility Study

2.2.1 Technical feasibility

The initial phase of the project involves the development of a client application with the primary goal of facilitating smooth connections with network nodes and peers. This client application will be the primary interface through which users will interact with the network. Furthermore, it will put emphasis on security by implementing strong mechanisms for securely keeping user data on the user's local device storage. This data storage option not only protects sensitive user information but also improves user control and privacy.

The proposed technological stack provides a strong foundation for reaching these goals. Using this stack, we can design and implement an application for clients that not only properly interfaces with network components but also smoothly integrates security measures into its architecture. This technical feasibility demonstrates the validity of our strategy, allowing us to design a client application that exactly corresponds with the project's objectives while maintaining a strong focus on data security and user experience.

2.2.2 Schedule Feasibility

Within the scheduled seven months, the solution must be implemented, with around two months allocated to requirement gathering and analysis and the remaining three months to system implementation. The system is tested during the duration of the final two months since testing is essential for the success of a project and allows for further improvements.

A requirement analysis was able to evaluate a timeline as well as the financial and technical feasibility because the research was carried out using a software-based methodology. This made it easier to match research goals, ensuring that they didn't go above available resources or timelines.

The primary goal of the literature review was to review the technique and materials used and to evaluate the conclusions that resulted from this field's study. Because this analysis provided accurate data, it contributed to the research work's decision-making process by contributing to determining the direction the research should take. A review of online resources relevant to already available monitoring solutions assisted in identifying the products' existing features and possible constraints.

2.2.3 Economic Feasibility

The initial phase of the project does not have any budgeted tasks since it focuses on the crucial requirement-gathering process, which proceeds through thorough evaluation and a review of past research and field studies. Throughout this period, the project efficiently uses open-source technology and tools, removing direct costs. However, as we progress through the development phase, costs must be expended. This phase requires the usage of cloud service provider services, which are essential for infrastructure and application development as well as scalability and flexibility. These expenses will be the primary focus of the thesis's next Economic Feasibility analysis, ensuring a detailed examination of the project's financial consequences.

2.3 Requirement Analysis

The requirement analysis phase was essential to this research project since it allowed for the identification of a number of components that should be considered during the research's implementation process.

During this phase, the data received from various sources during the requirement gathering phase was analyzed. As a result, critical factors relating to potential obstacles, as well as knowledge of the technique and an understanding of the use of potential tools and technologies, were easily discovered.

This step also allowed for the scope and the viability of the research to be determined. Furthermore, requirement analysis assisted in identifying present research gaps and provided insight into identifying the underlying research problem as the research progressed.

2.4 Software Development Life Cycle (SDLC)

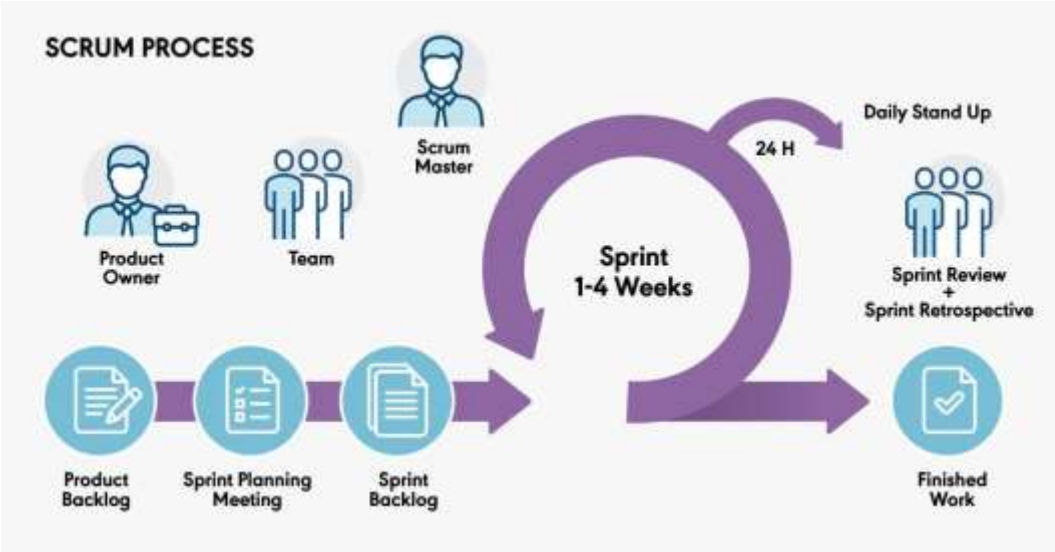

Figure 2.4.1: Software Development Life Cycle (SDLC)

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 [25]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 [25], SCRUM implies that the systems development process is unpredictable, complex and can only be described as an overall progression.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 measure a well-structured and systematic approach to achieving the research objectives.

2.5 System Architecture

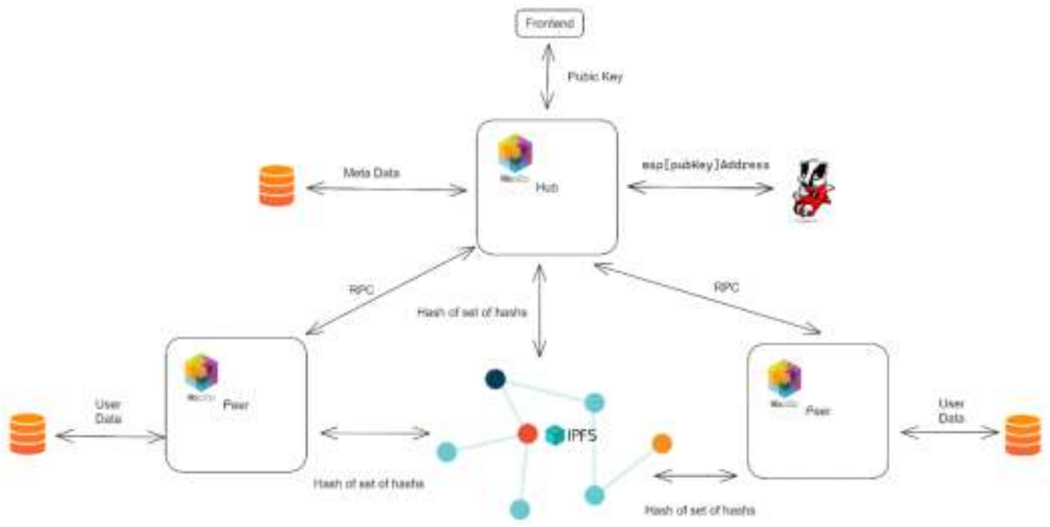

Figure 2.5.1: System Architecture (Overall)

The provided explanation outlines the components of the DeMedia architecture, which is a proposed infrastructure for a decentralized social media platform. The diagram accompanying the explanation provides a high-level overview of the architecture, which includes four major components: data decentralization protocol, peer-to-peer communication, decentralized data storage, and data integrity in a decentralized network. The first component of the DeMedia architecture is the data decentralization protocol, which will enable 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 will allow 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. 14 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 through the use of 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. The infrastructure will consist of hubs, peers, and a decentralized data storage network, which will be illustrated in the high-level architectural diagram.

2.5.1 System Architecture (Individual)

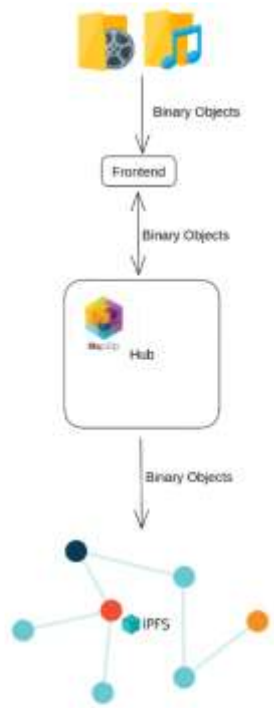

Figure 2.5.1.1: System Architecture – Individual

This component will produce a client application and a decentralized social media protocol with a caching system to show off its functionality. Performance, scalability, security, and usability of both the protocol and client application will be enhanced. They'll enable simple content sharing and distribution over the decentralized social media network, support a range of content categories, and support a number of different content models. There will be a user-friendly interface in the client application.

Our cost-effective method caches data using IPFS (Interplanetary File system), an alternative to costly blockchain technology. Users can easily submit content from our website, like MP3 audio files, which then move through our hub before securely existing in IPFS. The best thing is that we built it to be cost-effective, so users would experience no additional expenses. This efficient and cost-effective strategy ensures safe data storage and availability without affecting our customers' wallets.

2.6 Commercialization aspects of the product

DeMedia is focused on creating an open-source protocol that makes it easier to build self-hosted, decentralized social media sites. The project seeks to offer a free foundation concept that anyone interested in building a decentralized social media platform can use.

DeMedia will provide two paid models in addition to its free model: a membership-based subscription model and an advertising-based income model. The host can control these paid models, which enables them to make money on their site. DeMedia is therefore a research project with a particular interest in commercializing the creation of decentralized social media networks.

The project involves creating a variety of monetization mechanisms that hosts may use to make money from their platforms, allowing the technology to be commercialized.

DeMedia is focused on developing a technology that can support the creation of decentralized social media platforms, which is crucial to understand in order to further explain the monetization component of the project. DeMedia makes it simpler for people or organizations to develop their own social media platforms that are not governed by a single entity by offering a free base concept.

However, there must be a mechanism to make money in order to maintain and expand these platforms. Here's where the two premium models that DeMedia provides are useful. Hosts are able to charge customers for access to exclusive features or content on their platform because of the subscription-based membership model. This income can be utilized to pay for the platform's hosting and upkeep expenses.

The advertising-based income model, on the other hand, enables hosts to make money by showing adverts on their platform. Advertisers may be charged by hosts to have their adverts displayed on the platform, and this money may be used to pay for hosting and upkeep expenses as well as to make a profit.

As a summary of the proposed system, DeMedia enables the commercialization of decentralized social media platforms by supplying technology that enables anyone to develop their own platform as well as monetization approaches that enable hosts to make money and maintain their platforms. This might result in a more varied and decentralized social media ecosystem with additional platforms serving specialized markets 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 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.

The 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 us to meet project milestones and deliver robust outcomes.

2.7.3 Development

Within the scope of enhancing network performance and mitigating the issues associated with storing large media objects on the blockchain, the development efforts primarily focused on the implementation of a caching mechanism within the decentralized social media protocol. This caching mechanism leveraged the InterPlanetary File System (IPFS) to optimize data storage and retrieval while utilizing users' devices for data storage.

2.7.3.1 IPFS Integration for Media Object Caching

To address the challenges of storing large media objects within a decentralized social media protocol, the research component integrated the InterPlanetary File System (IPFS). IPFS is a distributed file system designed for efficient content addressing and retrieval. It enables the protocol to store media objects off-chain, reducing the burden on the blockchain network.

2.7.3.2 Data Caching for Network Performance

The key innovation of this research component lies in the implementation of a data caching mechanism on IPFS. When users upload media objects, these objects are cached on IPFS, allowing for quicker retrieval by other users. This caching mechanism significantly improves network performance by reducing the need to fetch large media objects directly from the blockchain.

2.7.3.3 User Device Utilization

A fundamental principle of the caching mechanism is the utilization of users' devices for storing cached media objects. Each user who interacts with the protocol contributes to the distributed storage of media objects on IPFS. This not only optimizes data retrieval but also distributes the storage load across the network, addressing concerns related to the cost and scalability of on-chain storage.

2.7.3.4 Cost-Effective Large Object Handling

By offloading the storage of large media objects to IPFS and distributing the storage load across the user network, the research component effectively addresses the challenges of cost and scalability associated with on-chain storage. Users collectively participate in maintaining the decentralized storage infrastructure, making it more cost-effective and sustainable.

2.7.3.5 Network Performance Enhancement

The caching mechanism implemented in this research component significantly enhances network performance. Users can access media objects more swiftly, reducing latency and improving the overall user experience. Furthermore, by minimizing the direct interaction with the blockchain for media retrieval, the protocol can scale more efficiently, accommodating a larger user base.

In conclusion, the development work for the research component focused on integrating IPFS for media object caching, utilizing user devices for storage, and enhancing network performance. In addition, these efforts help to overcome the challenges associated with storing and retrieving large media content within a decentralized social media protocol while enhancing efficiency and network performance.

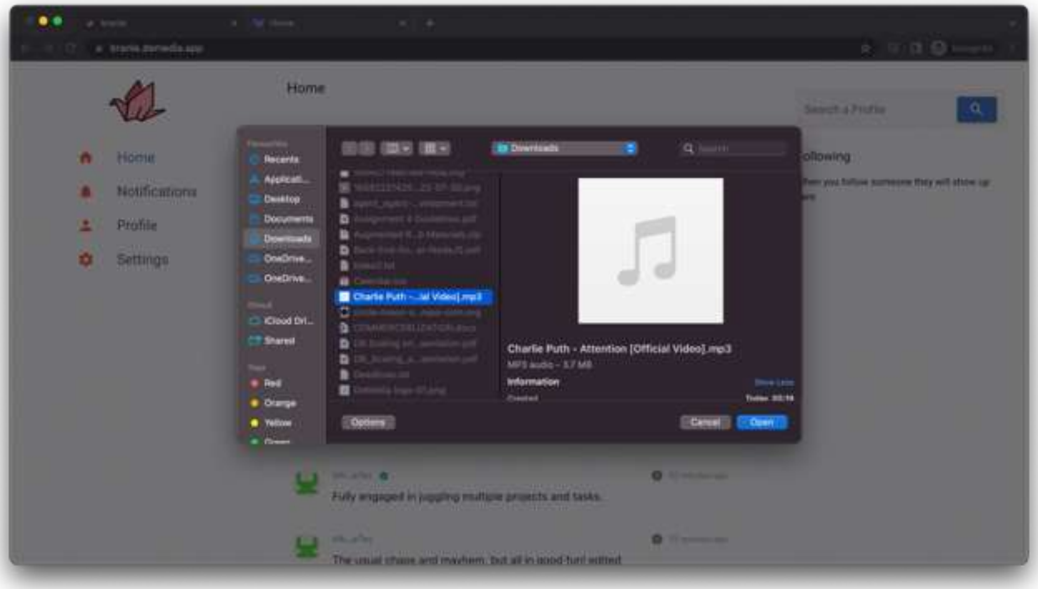

Figure 2.7.3.5.1: Upload the song from device

Figure 2.7.3.5.2: Playing Uploaded Song

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 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:

1. **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.
2. **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.
3. **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.

4. **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.

3.0 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.

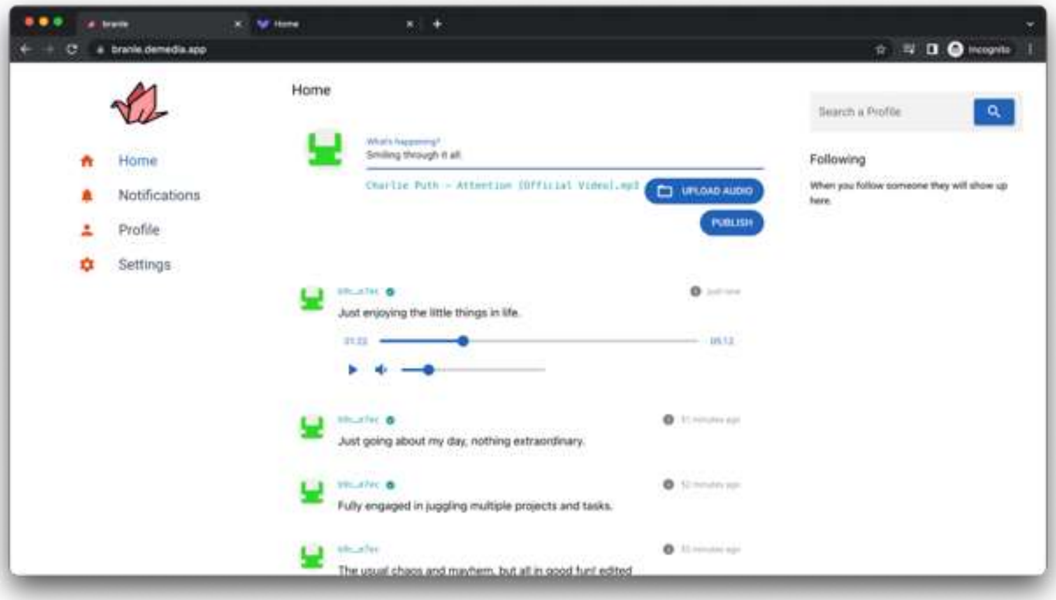

Figure 3.1.1.1: User upload mp3 from frontend

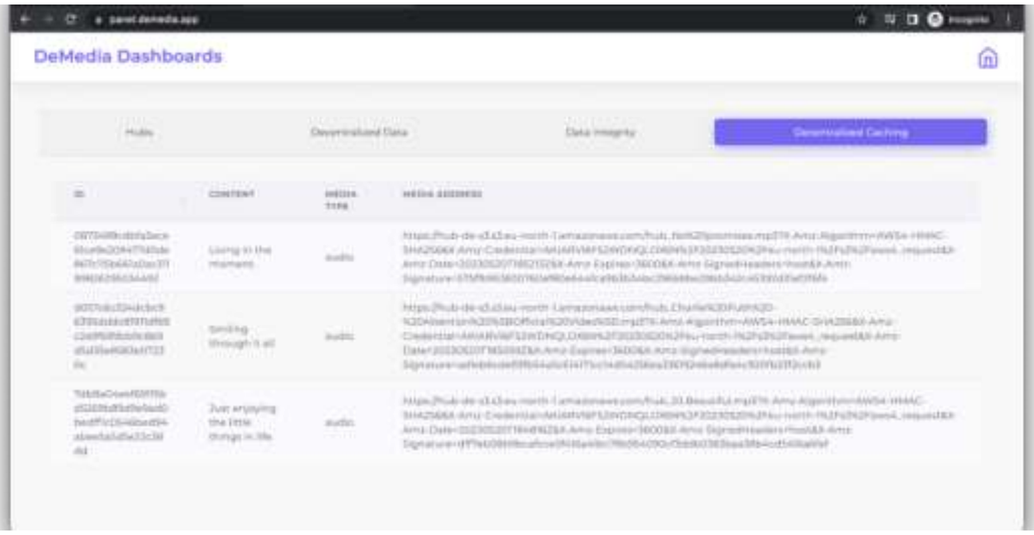

Figure 3.1.1.2: The panel of decentralized caching

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:

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 3.1.2.1: Output from benchmark application

3.1.1 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

Our research explored the practical use of caching mechanisms within a decentralized social media protocol, focusing on enhancing storage management and resource efficiency. We found that implementing caching significantly reduced data retrieval times, improving user experiences, and reducing latency.

It also optimized storage by minimizing redundancy through IPFS-based storage of frequently accessed data. Integrating this caching mechanism with IPFS offered efficient content updates and retrieval in a decentralized environment, reducing resource consumption. Our comprehensive documentation provides guidance for others looking to adopt and optimize similar solutions, contributing to better storage management and resource efficiency in decentralized social media protocols. In essence, our research highlights the potential of IPFS-based caching for improved data access and storage in decentralized social media platforms.

3.3 Discussion

Integrating the Interplanetary File System (IPFS) for media object storage was an important architectural aspect of the protocol. In the demo platform, media objects were cached on IPFS, allowing users to access and share content efficiently. This caching mechanism showcased the protocol's ability to leverage decentralized data storage while ensuring seamless content retrieval. The media object caching capability shows the protocol's scalability and potential to reduce demand for network resources and control the cache handed over to the user.

Figure 3.3.1: Listening to the uploaded song

4.0 SUMMARY OF EACH STUDENT’S CONTRIBUTION

Student Number	Name	Tasks
IT20157432	Abeykoon A.W.Y.I.K.	• Conduct a comprehensive review of data caching mechanisms and associated research. • Assess the opportunities and challenges present in existing data caching mechanisms. • Determine the potential enhancements and new capabilities that could be integrated into a mechanism for data caching in a peer-to-peer network. • Implement a mechanism to cache data in a peer-to peer network with the identified modifications while adhering to best practices.

Table 4.1: Student's contribution

5.0 CONCLUSION

In conclusion, the decentralized social media protocol is rapidly expanding, and several protocols have taken aim for control. However, as we've previously mentioned, many of these protocols face significant challenges, especially in terms of scalability and data storage.

In order to address these problems, IPFS (Interplanetary File System) appears as an important solution. With its innovative, decentralized technology for data storage, it transforms online content management. Its caching method is an essential strength since it improves protection and scalability by effectively handling frequently accessed data, which leads to a better user experience.

What defines IPFS is its constant focus on decentralization. Unlike traditional server-based data storage, IPFS distributes and stores things via a worldwide network of nodes. This technique not only improves data adaptability but also reduces censorship concerns, which is essential in the context of social media where the flow of information is essential.

IPFS becomes more essential as the decentralized social media system grows and gets acknowledgement. Its efficient and secure data storing characteristics make it an important technology for social media's future. Furthermore, IPFS's effective, open-source nature combines well with the ideas of decentralization and user empowerment, assuring that it will continue to play an important role in determining the future of social media. Against a background of increasing challenges and opportunities, IPFS stands as an example of transparency, accessibility, and adaptability in this dynamic environment.

REFERENCES

[1] M. Dollarhide, "Social Media: Definition, Importance, Top Websites & Apps," 31 August 2023. [Online]. Available: https://www.investopedia.com/terms/s/social-media.asp. [Accessed 06 September 2023].

[2] "The evolution of social media: How did it begin and where could it go next?," Maryville Online, 03 March 2021. [Online]. Available: Maryville Online. [Accessed 06 September 2023].

[3] C.-m. A. Yeung, I. Liccardi, K. Lu, O. Seneviratne and T. Berners-Lee, "Linking the World’s Information," Decentralization: The future of online social networking, p. 12, 2023.

[4] "What is web3 and why is it important?," 01 September 2023. [Online]. Available: https://ethereum.org/en/web3/. [Accessed 07 September 2023].

[5] N. Dabit, "What is web3? the decentralized internet of the future explained," 26 September 2021. [Online]. Available: https://www.freecodecamp.org/news/what-is-web3/. [Accessed 07 September 2023].

[6] L. Schwittmann, M. Wander, C. Boelmann and T. Weis, "Privacy Preservation in Decentralized Online Social Networks," IEEE Internet Computing, 2013.

[7] T. Sarathchandra and D. Jayawikrama, "A decentralized social network architecture," 2021 International Research Conference on Smart Computing and Systems Engineering (SCSE), pp. 251-257, 2021.

[8] S. Shilina, "What are decentralized social networks?," 04 July 2023. [Online]. Available: https://cointelegraph.com/explained/what-are-decentralized-social-networks#:~:text=Hive%20is%20a%20decentralized%20social,free%20speech. [Accessed 07 September 2023].

[9] C. Products, "Peepeth," 04 February 2020. [Online]. Available: https://cryptoslate.com/products/peepeth/. [Accessed 07 September 2023].

[10] "Mastodon (Social Network)," 29 August 2023. [Online]. Available: https://en.wikipedia.org/wiki/Mastodon_(social_network). [Accessed 07

September 2023].

[11] "The Blockchain & Cryptocurrency for Web3," [Online]. Available: https://hive.io/. [Accessed 07 September 2023].

[12] "Steemit," 10 June 2023. [Online]. Available: https://en.wikipedia.org/wiki/Steemit. [Accessed 07 September 2023].

[13] "LevelDB," 21 November 2018. [Online]. Available: https://chenju2k6.github.io/blog/2018/11/leveldb. [Accessed 07 September 2023].

[14] R. Sheldon, "7 decentralized Data Storage Networks compared: TechTarget," 17 August 2023. [Online]. Available: https://www.techtarget.com/searchstorage/tip/Comparing-4-decentralized-data-storage-offerings. [Accessed 07 September 2023].

[15] S. Williams, V. Diordiiev, L. Berman, I. Raybould and I. Uemlianin, "ARWEAVE: A protocol for economically sustainable information permanence," [Online]. Available: https://www.arweave.org/yellow-paper.pdf. [Accessed 07 September 2023].

[16] E. D. a. F. Tschorsch, "IPFS and Friends: A Qualitative Comparison of Next," IEEE Communications Surveys &, 2022.

[17] I. S. Labs, "Storj: A Decentralized Cloud Storage NetworkFramework," October 30, 2018.

[18] I. E. A. K. A.-A. J. I. S. H. S. S. U. N. M. I. Khalid, "A Comprehensive Survey on Blockchain-Based Decentralized Storage," vol. 11, pp. 10995-11015, 2023 January 27 .

[19] "What is IPFS," [Online]. Available: https://docs.ipfs.tech/concepts/what-is-ipfs/. [Accessed 07 September 2023].

[20] A. M. a. L. R. B. Guidi, "Data Persistence in Decentralized Social Applications: The IPFS approach," IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2021.

[21] "Advantages and disadvantages of decentralized blockchains," World Crypto Index, [Online]. Available: https://www.worldcryptoindex.com/advantages-

disadvantages-decentralized-blockchains/. [Accessed 08 September 2023].

[22] "Pin files using IPFS," IPFS Docs, [Online]. Available: https://docs.ipfs.tech/how-to/pin-files/. [Accessed 08 September 2023].

[23] G. Iredale, "Top 10 blockchain adoption challenges," 101 Blockchains, 15 August 2022. [Online]. Available: https://101blockchains.com/blockchain-adoption-challenges/. [Accessed 08 September 2023].

[24] M. Brown, "5 challenges with blockchain adoption and how to avoid them: TechTarget," TechTarget, 27 May 2021. [Online]. Available: https://www.techtarget.com/searchcio/tip/5-challenges-with-blockchain-adoption-and-how-to-avoid-them. [Accessed 08 September 2023].

[25] R. Keefe, "5 popular project management methodologies and when to use them," Toggl Blog, 13 September 2022. [Online]. Available: https://toggl.com/blog/5-popular-project-management-methodologies-and-when-to-use-them. [Accessed 10 September 2023].

GLOSSARY

- Caching: the process of storing data in a temporary storage location for quick access and retrieval.
- Blockchain: A decentralized, distributed digital ledger that records transactions across multiple devices and networks.
- Distributed caching: a caching mechanism in which data is stored across multiple nodes in a decentralized network.
- Content-addressable storage: a mechanism in which data is stored based on its content rather than its location, allowing for efficient retrieval and distribution.
- Content Delivery Network (CDN): a distributed caching system that stores and delivers content to end-users based on their geographic location, reducing latency, and improving performance.
- Peer-to-peer (P2P) caching: a decentralized caching mechanism in which nodes in a network cache data for each other, reducing the load on the network and improving performance.
- Blockchain: Transparency and immutability are guaranteed by a distributed database that keeps records of all data changes and transactions within the decentralized network.
- Node: A device or computer operating in a decentralized network that manages and saves user data snippets, enhancing the adaptability and availability of the network as an entire system.

APPENDICES

Appendix A : Turnitin Paper

Final Thesis

ORIGINALITY REPORT

5%	3%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	2%
2	Submitted to University of Ghana Student Paper	1%
3	Submitted to University of Hertfordshire Student Paper	<1%
4	www.researchgate.net Internet Source	<1%
5	Submitted to Lebanese American University Student Paper	<1%
6	5dok.net Internet Source	<1%
7	deasilex.com Internet Source	<1%
8	Submitted to CSU, Fullerton Student Paper	<1%
9	www.emo.org.tr Internet Source	<1%

Appendix A: Turnitin Report