PRELIMINARY PROJECT REPORT

ON

ONLINE VIDEO/AUDIO STREAMING SERVICE BASED ON DECENTRALISED ARCHITECTURE

Submitted by

ISHAN JOSHI

KISHLAYA KUNJ

NEERAJ LAGWANKAR

In partial fulfillment for the award of the degree

Of

Bachelor of Engineering

Of

Savitribai Phule Pune University

IN

INFORMATION TECHNOLOGY

MIT- COLLEGE OF ENGINEERING

Pune, Maharashtra, India

2018-19

PRELIMINARY PROJECT REPORT

ON

ONLINE VIDEO AND AUDIO STREAMING SERVICE BASED ON DECENTRALIZED ARCHITECTURE

Submitted by

ISHAN JOSHI

KISHLAYA KUNJ

NEERAJ LAGWANKAR

Guided by

Prof Shamla Mantri

DEPARTMENT OF INFORMATION TECHNOLOGY

MIT- COLLEGE OF ENGINEERING

PUNE

SAVITRIBAI PHULE PUNE UNIVERSITY

2018-19

Department of Information Technology

Certificate

This is to certify that,

T150388574 - Ishan Joshi

T150388594 - Kishlaya Kunj

T150388606 – Neeraj Lagwankar

Have successfully completed this project report entitled "**Online Video and Audio Streaming Service Based on Decentralized Architecture"** ,under my guidance in partial fulfillment of the requirement for the degree of Bachelor of Engineering in Department of Information Technology of Savitribai Phule Pune University, Pune during the academic year 2018-19

Date:

Pune

Prof Shamla Mantri
Guide

Insert name here
Head of the Department

ACKNOWLEDGEMENT

We take this opportunity to thank our project guide Shamla Mantri and Head of the Department <Name of Head> for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of MIT College of Engineering, Pune for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, internet access and important books.

Ishan Joshi
Kishlaya Kunj
Neeraj Lagwankar

CONTENTS

INTRODUCTION

1.1. Need

A. Lack of Robustness

Servers can be easily taken down using attacks such as Distributed Denial of Service (DDoS) attack, which is the most commonly used method. Such attacks exploit the fact that legitimate users can be denied access to the server/service by flooding the server with multiple requests. Such attacks would turn out to be useless against Web 3.0. Since, the data is never stored at one location/machine, the attack can be made only on a single system, which can be detected and necessary actions can be taken, without affecting the network.

B. Security

The decentralized nature of data also makes the network much more secure. This is due to the fact that current exploits would be rendered useless, due to the group of technologies constituting Web 3.0. Thus, manipulating data, or illegal access of data would be impossible, making the system secure, and free of plagiarized content.

C. Concurrent Content Delivery

It is often observed that servers go down due to the fact that multiple users try to access a given data stream. Upgradation of server is an expensive solution to the above stated problem. In our proposed system, the number of users will only strengthen the network, instead of degrading it. Video and audio streaming service using decentralized architecture which is based on technologies such as Peer to Peer networks, Merkle DAGs, and, Blockchain, would reduce the cost of service since there would be no need of renting, upgrading or maintaining servers. It would provide media content faster to end user. The inherent security of the network would also render conventional exploits useless, adding to the robustness of the system. Thus, providing a fast, secure and robust decentralized app (dapp) for users.

1.2 Applications

- It has a large potential to transform business operating models in the long term.
- The Government of India is fighting land fraud with the help of blockchain.
- Counterfeit of original documents can be prevented.
- Network load on servers can be removed and faster access to files will be facilitated.
- Blockchain-based smart contracts are contracts that can be partially or fully executed or enforced without human interaction.
- The advent of blockchain has resulted into many things like a decentralized file system, decentralized library, etc.
- It is being extensively used by financial institutions for crypto currency and data security.

LITERARTURE SURVEY

A decentralized web will be able to solve the problems faced by any service based on client server architecture. In the last decade, advancement in web technology has led to the concept of decentralized network, thus allowing the rise of peer-to-peer communication. Peer-to-Peer communication circumvents this problem by relaying traffic through peers instead of a dedicated server.

The components of the decentralized web include Peer to Peer (P2P) file sharing, Distributed Hash Tables (DHT), blockchain, Self-certifying File Systems, Consensus Protocols and Smart Contracts.

A. Peer to Peer File Sharing

Applications of P2P file sharing such as BitTorrent leverage its users' resources to distribute all types of digital files to its consumer without the need of a central governing model. Client server architecture based content sharing services often incur high electricity cost to maintain high speeds of content delivery, to maintain the temperature of the servers among many other factors . Since P2P architecture is self-maintaining, resilient and only need limited infrastructure and control, it is vastly superior, faster, more secure and robust than the existing client server architecture.

B. Distributed Hash Table

Distributed Hash Tables are used as a lookup service in distributed and decentralized services. They are used to map identifiers from a common pool of peers or nodes in an overlay network. A DHT is an extension of a simple hash table that saves data in the form of key-value pairs on Node IDs. The Node Id is generated using the nodes' IP address or geographic information. The key is generated using a custom hash function using the data item as a parameter. Most of the existing DHT assume that its peers are spread over the ID space uniformly.

C. Blockchain

Blockchain is a distributed, transparent, immutable ledger having a consensus protocol at the root of it. It is a growing database of records that have been executed among peers who have taken part in the transaction. These ledgers are visible and using a P2P approach, the peers or nodes in the blockchain network can edit the distributed ledger. This makes tampering with the blocks comprised within the Blockchain extremely challenging given the cryptographic data structure used in blockchain and no necessity for secrets. Simply said, a blockchain is analogous to a singly linked list, where, instead of a pointer to the next node, we have a hash of the current block and the previous. A node or a block contains a timestamp, a set of transactions, a nonce, hash of the current block and hash of its predecessor.

Working of a blockchain network:

1. Peers interact with the blockchain network using asymmetric encryption. Asymmetric Encryption uses two different keys - public and private keys to encrypt and decrypt data. Nodes/peers use private key to digitally sign their own transactions and are addressable on the network by public key. Every transaction is broadcasted by a node in the network.
2. This transaction is then validated by all the nodes in the network barring the one which created it. Invalid transactions are discarded. This process is called verification.
3. Each node collects the transactions that have been validated in a certain time into a block and implements a proof-of-work finding a nonce for its block. When a node finds a nonce, it broadcasts the block to all nodes. This is a process called mining.
4. All nodes select a block broadcasted for the first time and verify that the block
   (a) contains valid transactions and
   (b) references via hash the correct previous block on their chain.

If that is the case, they add the block to their chain and apply the transactions it contains to update their blockchain. If that is not the case, the proposed block is discarded. This marks the end of current mining round. Blockchains can be classified into three major categories - public blockchain, consortium blockchain and fully private blockchain. A public blockchain is openly available and every peer or node has an equal right to validation of a transaction. A consortium blockchain, too is openly available, but in this case, each node can have different rights of validation of transactions. A private blockchain is not openly available. In a private blockchain, a single, centralized governing authority has the right to validate a transaction.

The figure below depicts a blockchain.

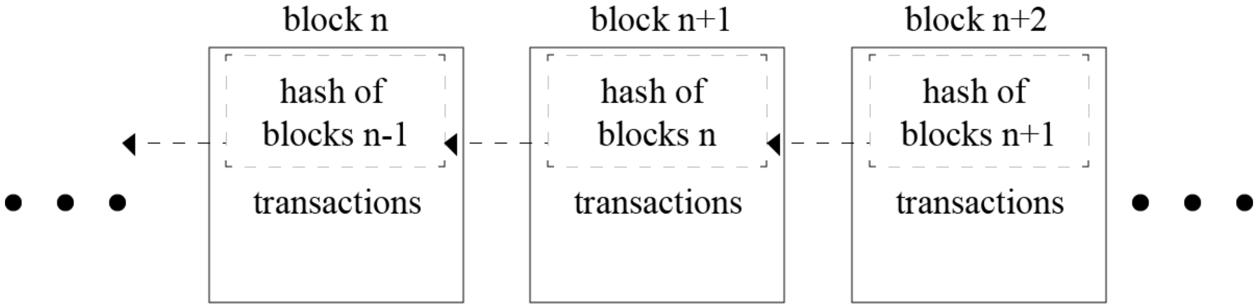


Fig. 1. Blockchain

D. Self-certifying File System

SFS is a secure file system spread over the internet. It provides one namespace for all the files in the world. SFS is inherently secure, and hence its users can share sensitive files without worrying about a third party tampering or reading the file. IPFS uses the public key cryptography used in SFS.

E. Consensus Protocols

Consensus Protocols are the backbone of any blockchain application. Such protocols are used to provide authenticity, non-repudiation and integrity to the blockchain network, by utilizing a decentralized peer-to-peer network for verification of transactions before adding a block to the public ledger. The bitcoin blockchain uses the concept of Proof of Work (PoW) to help decide validate the transactions occurring and also helps in avoiding the forking problem in blockchain. Some other types of Consensus Protocols are Delegated Proof of Stake (DPoS), Proof of Activity (PoA) - an amalgamation of PoW and PoS.

F. Smart Contracts

Paper contracts take a lot of time to travel around the globe and digital documents are relatively easier to forge. In order to automate the transactions to make them smoother, more efficient, and more secure for all the clients, smart contracts are coming in the scenario. Smart contracts were first proposed by Nick Szabo in the early 1990s. A smart contract is “a digital contract that is written in source code and executed by computers, which integrates the tamper-proof mechanism of blockchain”. Smart contracts have transformed the blockchain scenario from a financial transaction protocol to an all-purpose utility. They are pieces of software, not contracts in the legal sense, that extend blockchain’s utility from maintaining a ledger of financial transactions to automatically implementing conditions of multi-party agreements. Smart contracts are executed by a computer network that uses consensus protocols to agree upon the series of actions resulting from the contracts content . The high-level programming languages used for writing smart contracts are mainly Solidity, Serpent and Low-level Lisp-like Language (LLL).

G. Git

Technological growth has happened at a very high rate in the recent decades, especially in the field of computers. Computers have evolved from huge ineffective mainframe computers to today's portable highly effective laptops, mobile phones and desktops. Software being an integral part of the computer system, large number of project files are created. To

keep track of all the changes in the files, a version control system is used. One of the most popular version control system is Git. One of the advantages for using Git is being open source in nature, and data can be extracted easily through the change logs maintained by it. Git maintains huge number of open source software systems. Version control system allows multiple users to store changes and switch and access different versions with ease. Master branch is the current working branch that is the most stable branch. One can commit the changes to the master branch if the number of changes are small and does not significantly affect the working of the program. However the branch must be forked to incorporate some new changes which change the working of the program altogether. This ensures that the master branch is not affected and work can be done in the newly created development branch. If the user finds that the newly created branch is stable, it is upto the user to merge the development branch and the master branch to incorporate the changes in the master branch. This extremely flexible branching structure enables developers not only to increase productivity, but also handle various development activities. Figure 2 depicts the forking and merging of branches.

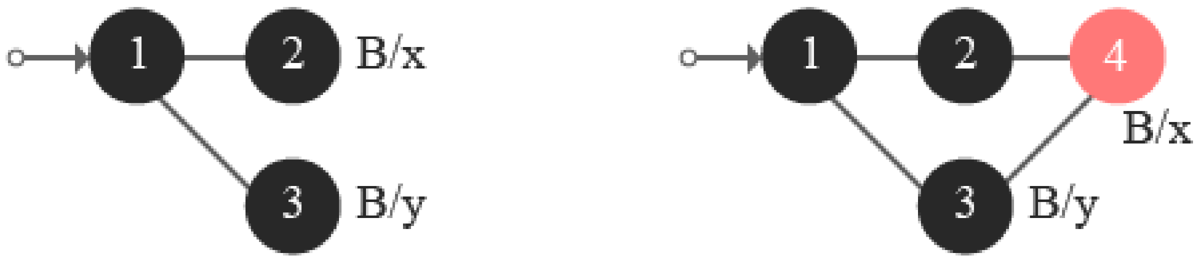


Fig. 2. Forking and merging of Git branches

III. EXISTING TECHNOLOGIES

The blockchain is a relatively new approach in the field of information technology. The complexity of the technology poses many challenges and foremost amongst these are management and monitoring of blockchain based decentralized applications. Next section takes a deep dive in two such technologies - Ethereum and IPFS. A. Ethereum

Ethereum, proposed by a cryptocurrency researcher and programmer Vitalik Buterin, is a public, open-sourced, blockchain-based distributed computing platform having smart contract functionality. The first live Ethereum blockchain network was launched in 2015. Ethereum cryptocurrency token "Ether" as of July 2018, third in popularity after Bitcoin and Ripple, is used to compensate participating peers for computations performed . Ethereum represents a blockchain with a built-in Turing complete programming language called Solidity. It facilitates an abstract layer allowing anyone to create their own rules for ownership, formats of transactions, and state transaction

functions. This is achieved by inculcating smart contracts, a set of cryptographic rules that are processed only if all necessary conditions are satisfied. The consensus in the Ethereum network is based on modified GHOST protocol (Greedy Heaviest Observed Subtree). It is created to solve the issue of stale blocks in the network. If one group of miners combined in a mining pool has more processing power than the others, it results in formation of stale blocks. The blocks from the first pool will contribute more to the network which in turn creates the centralization issue. GHOST protocol includes those stale blocks into calculations of the longest chain.

B. InterPlanetary File System

The InterPlanetary File System (IPFS) is a distributed file system which incorporates ideas from existing technologies like BitTorrent, Git, SFS, and Kademlia and models them into a complete system. IPFS, is a peer-to-peer distributed file system, aims to replace HTTP and build a better web for us all . The torrent protocol facilitates relocation of data between nodes comprising the infrastructure and the Kademlia DHT is used for the management of metadata. IPFS can be assumed as a single BitTorrent collection which exchanges data within one Git repository. IPFS facilitates a high-put content-addressed block storage model, with content addressed hyperlinks. IPFS includes a distributed hashable, reward-driven block exchange, and a self-certifying namespace. IPFS doesn't have more than one point of failure, and it is not mandatory for the peers to trust one another. IPFS borrows the concept of Merkle Directed Acyclic Graphs (DAGs) from the Git Version Control System. The Merkle DAG object model helps capture changes to the IPFS tree, or even a permanent web, in a distributed-friendly way. Figure 3 depicts the creation of a new object: the client sends its object to any node on its nearest site.

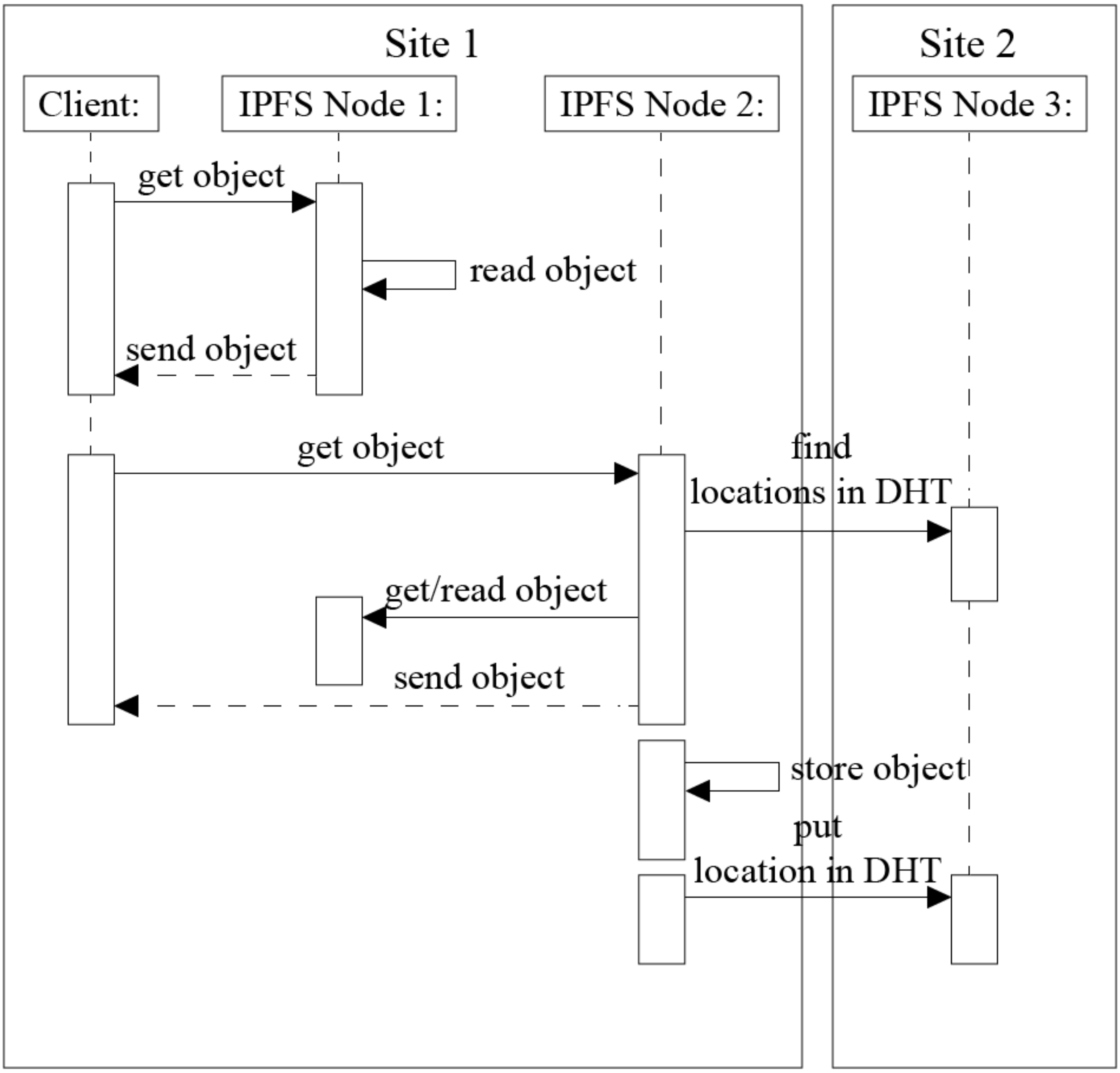


Fig. 3. Reading an IPFS Block

This node stores the object locally and put the location of the object in the DHT. Because the DHT does not provide locality, the node storing this metadata can be located in any node composing the Fog infrastructure. In our example, Node 4 belonging to Site 2 stores the location of the object that has been created on Node 1. Figure 4 illustrates what happens when the client reads an object stored on its local site.

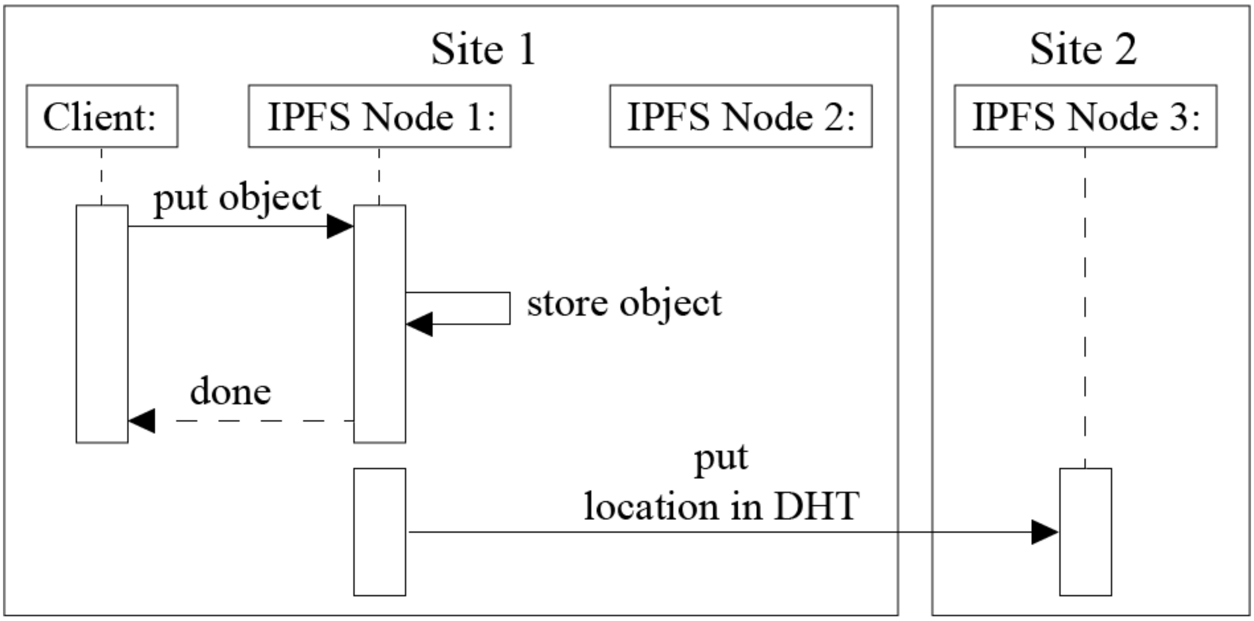


Fig. 4. Writing a block to IPFS

. Each time an IPFS node receives a request for a particular object, it first, checks whether this object is available on the node. In this case, the node sends the object directly to the client. Otherwise, the IPFS node should rely on the DHT protocol to locate the requested object. That is, it should compute the hash based on the object id, contact the node in charge of the metadata, retrieve the object from the node(s) storing it (using the BitTorrent protocol), make a local copy while sending the data to the client, and finally update the DHT in order to inform that there is a new replica of the object available on that node. Figure 5 describes what happens when an object is requested from another site (because the client moves from a site to another one or because the object is accessed by a remote client).

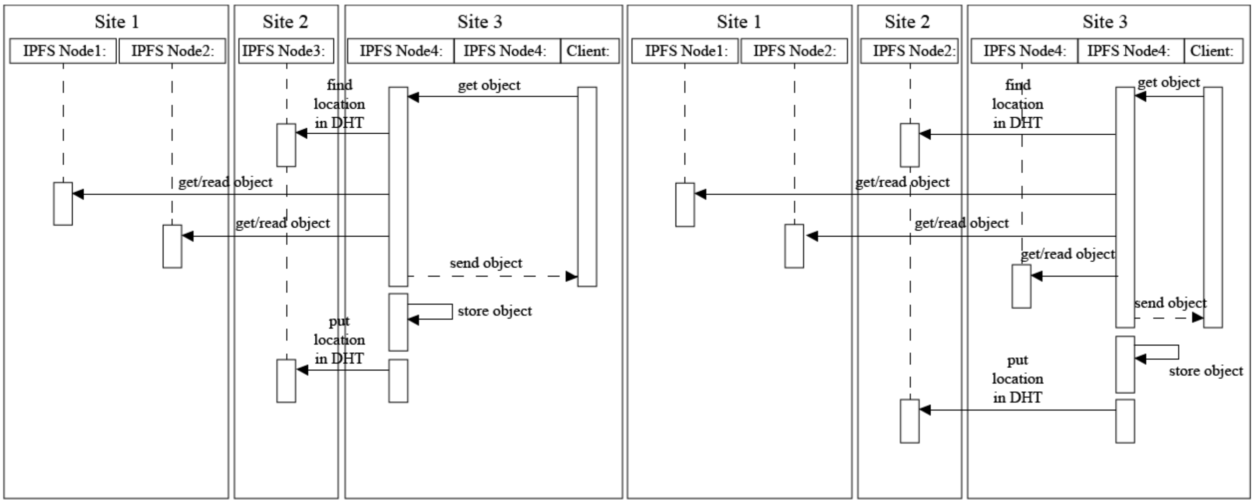


FIG. 5. Read an object stored remotely

In any case, the client contacts a node on the site it belongs to. Because the node does not store the object, the protocol is similar to the previous one involving the extra communication with the DHT. In the network model of decentralized services, applications distribute their workload over multiple hosts . In some cases, the DApp is composed of components, each running on any random and independent host that provides the requisite services. It is very complex to manage the decentralized network. The administration of the network is also very complex. A problem of these approaches arises when a node fails to provide the desired service. In this case, the network has to look for the service in another node. If it still cannot find the node with the desired service, it will keep on looking in different nodes. If the requested service is not found, there has to be some kind of timeout agreement in order to prevent the user having to wait endlessly. The problem with system implemented with peers is that no one is responsible. If someone uploads a family picture, and ten years later, he wants the photo that is stored in the decentralized network, all those nodes may have been gone. The data might have been erased unknowingly years before.

2.1 Related Works

2.1.1 Journal Paper

1. Donhee Han, Hongjin Kim, Juwook Jang, “Blockchain based Smart Door Lock System”, IEEE, Information and Communication, pp. 1165, 2017.
2. Robert W. Lucky, "The Lure of Decentraisation", IEEE Spectrum, pp. 23, 2017.
3. Y. Sompolinsky and A. Zohar, “Secure high-rate transaction processing in Bitcoin,”, Springer, Financial Cryptography, pp. 507-527, 2015.
4. Konstantinos Christidis, and, Michael Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things”, IEEE Access, Special Section on the Plethora of Research in Internet of Things, 2016.

2.1.2 Conference Paper

1. Malik Muhammad Imran Pattal, Li Yuan, Zeng Jianqiu, “Web 3.0: A real personal Web”, IEEE, Third International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 125128, 2009

2. Lakshmi Siva Sankar, Sindhu M, M. Sethumadhavan, “Survey of Consensus Protocols on Blockchain Applications”, IEEE, International Conference on Advanced Computing and Communication Systems, 2017.
3. Ruchika Malhotra, Nakul Pritam, Kanishk Nagpal, "Defect Collection and Reporting System for Git based Open Source Software", IEEE, International Conference on Data Mining and Intelligent Computing (ICDMIC) 2014
4. Jiin-Chiou Cheng, Narn-Yih Lee , Chien Chi , and Yi-Hua Chen, “Blockchain and Smart Contract for Digital Certificate”, IEEE, Proceedings of IEEE International Conference on Applied System Innovation, 2018.
5. Fabius Klemm, Sarunas Girdzijauskas, Jean-Yves Le Boudec, Karl Aberer, “On Routing in Distributed Hash Tables”, IEEE, 7th International Conference on Peer-to-Peer Computing, pp. 113-120, 2007.
6. Deepak K. Tosh, Sachin Shetty, Xueping Liang, “Consensus Protocols for Blockchain-based Data Provenance: Challenges and Opportunities”, IEEE, 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), pages 469–474, 2017.
7. Sachchidanand Singh, Nirmala Singh, “Blockchain: Future of Financial and Cyber Security”, IEEE, 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 463–467, 2016.
8. Dejan Vujičić, Dijana Jagodić, Siniša Ranđić , “Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview”, IEEE, 17th International Symposium INFOTEH-JAHORINA, 2018.

2.1.3 Study Papers

1. Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, White Paper, 2008.
2. V. Buterin, “Ethereum white paper: a next generation smart contract & decentralized application platform,” Ethereum White Paper, 2013.
3. Smith, Jerry. “Distributed Computing with Aglets”. White Paper, 1999.
4.
5. J. Blackburn, K. Christensen, ”A Simulation Study of a New Green BitTorrent,” IEEE, Proc. Green Communications Workshop in conjunction with IEEE ICC’09, 2009.

6. Olaf Landsiedel, Stefan Gotz, Klaus Wehrle, "Towards Scalable Mobility in Distributed Hash Tables" IEEE, Peer-to-Peer Computing, 2006.
7. David Mazieres, "Self Certifying File System", Doctor of Philosophy, Massachusetts Institute of Technology, Massachusetts, USA, 2000.
8. Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System (Draft 3)", White Paper, 2014.
9. Konstantinos Christidis, and, Michael Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things", IEEE Access, Special Section on the Plethora of Research in Internet of Things, 2016.

PROJECT STATEMENT

Today, in this rapidly changing world, media consumption is not a luxury anymore, it is a necessity. People view millions of videos every minute through various websites such as YouTube, Dailymotion and Twitch. It is thus, important that the user is provided with the desired content in a fast and timely fashion, with limited packet loss. Thus, we are trying to implement blockchain in video and audio streaming services which overcomes the drawbacks of traditional services. Blockchain aims at decentralizing the web in order to remove the middlemen (the servers) to provide peer to peer connectivity between users. It is considered as the future of internet. We hope to develop such a system that is able to deliver media content such as video or audio without delay or traffic issue. We aim to use modern technologies such as Blockchain, Proof of Stake and merging them with older technologies such as Peer to Peer networks/Adhoc systems, to develop a robust, fast and secure service platform.

HARDWARE AND SOFTWARE REQUIREMENTS

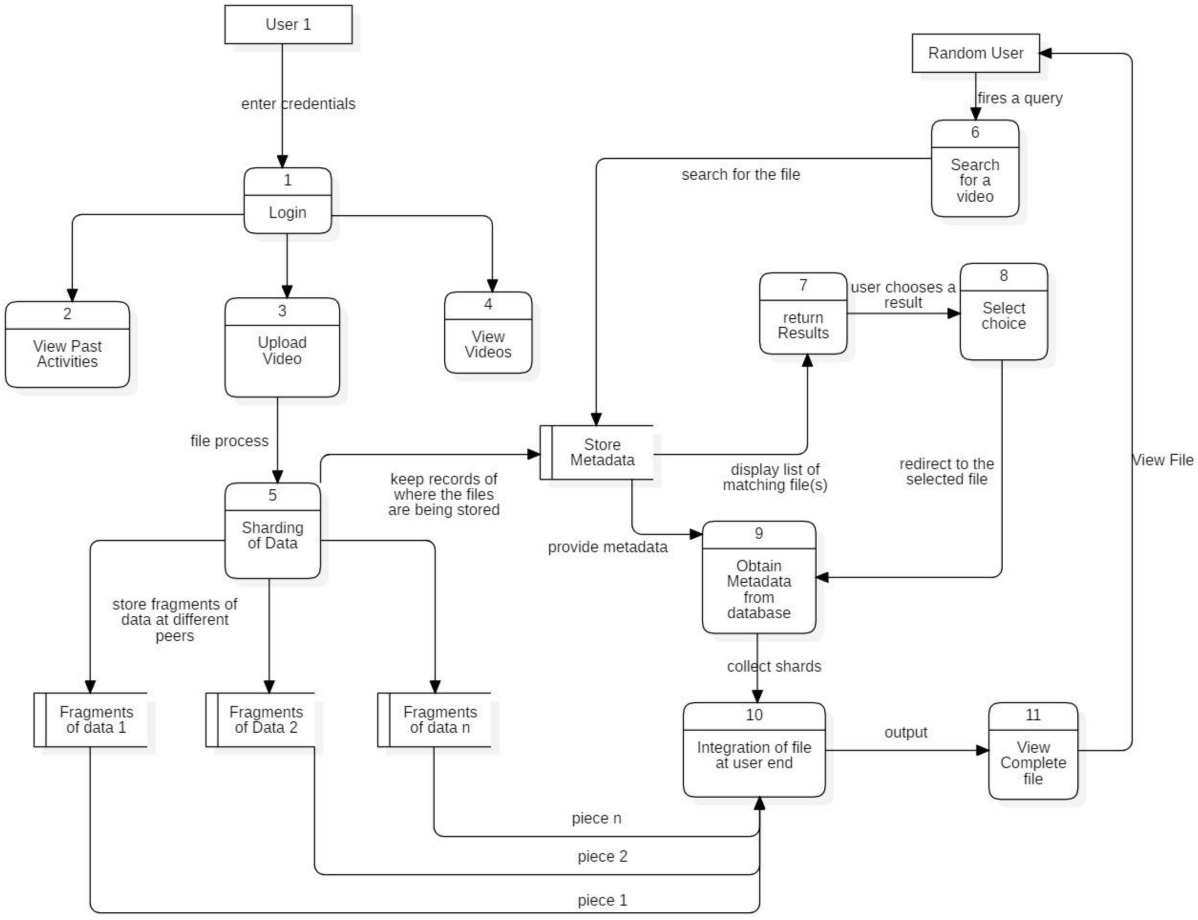
4.1 Hardware

- **Processor:** Intel Core 2 Duo or later.
- **Memory:** 2 GB RAM.
- **Graphics:** Video card must be 256 MB or more and should be a DirectX 9-compatible.
- **Storage:** 2 GB available space

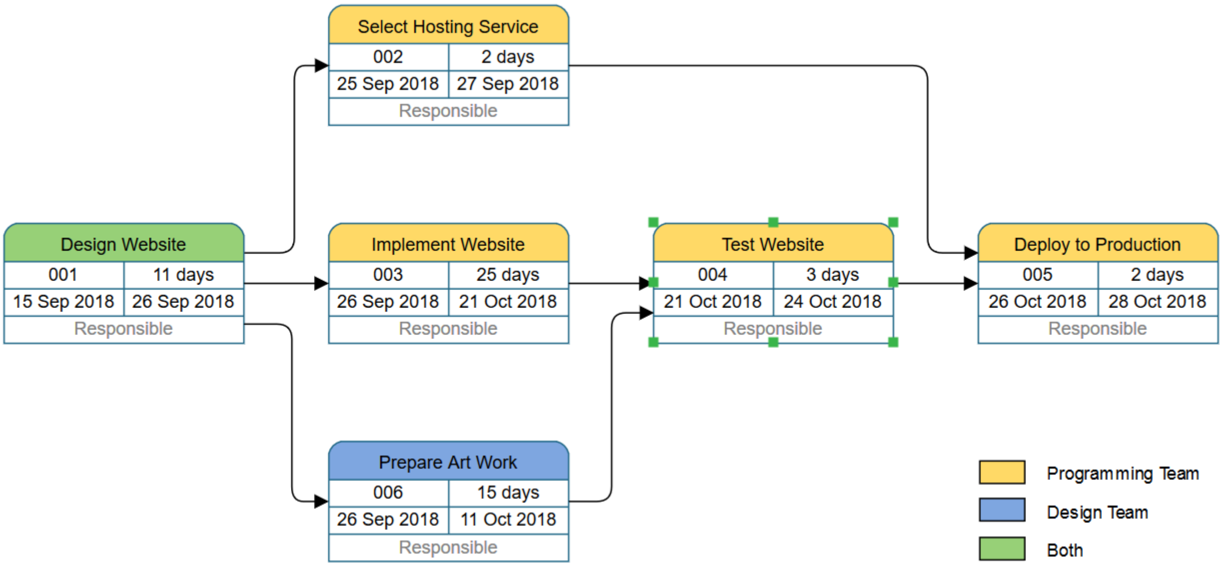
4.2 Software

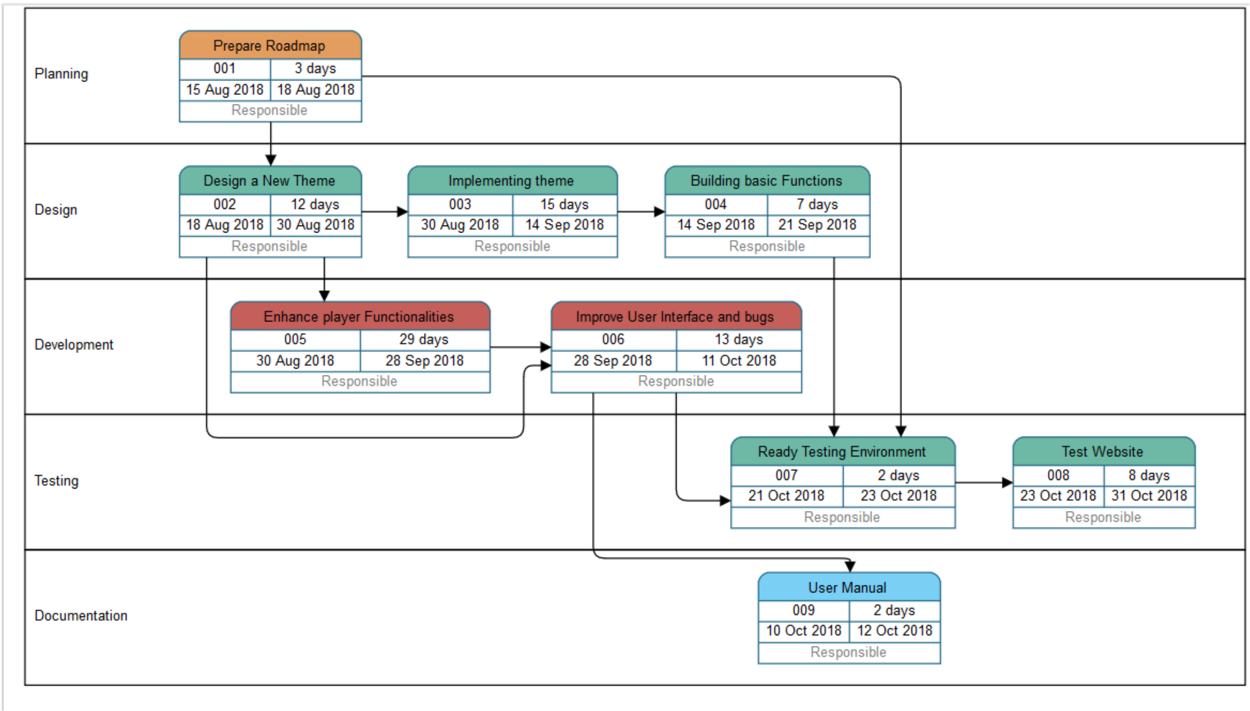
- **Operating System:** Windows 7/8/10, MacOS, any Linux based OS
- **Browsers** : Mozilla Firefox, Google Chrome
- IPFS

DATA FLOW DIAGRAM


User 1
enter credentials
1
Login
2
View Past Activities
3
Upload Video
4
View Videos
file process
5
Sharding of Data
keep records of where the files are being stored
store fragments of data at different peers
Fragments of data 1
Fragments of Data 2
Fragments of data n
piece n
piece 2
piece 1
Random User
fires a query
6
Search for a video
search for the file
7
return Results
user chooses a result
8
Select choice
Store Metadata
display list of matching file(s)
redirect to the selected file
View File
provide metadata
9
Obtain Metadata from database
collect shards
10
Integration of file at user end
output
11
View Complete file

PERT





LIST OF FIGURES

LIST OF TABLES