InterPlanetary File System and Ethereum: A Brief Overview

Kishlaya Kunj
Dept. of Information Technology
MIT College of Engineering
Pune, India
kishlayakunj@gmail.com

Neeraj Lagwankar
Dept. of Information Technology
MIT College of Engineering
Pune, India
neerajlagwankar@gmail.com

Ishan Joshi
Dept. of Information Technology
MIT College of Engineering
Pune, India
ishanjoshi0209@gmail.com

Shamla Mantri
Dept. of Information Technology
MIT College of Engineering
Pune, India
shamla.mantri@mitcoe.edu.in

Abstract — **In today's age, most of the services are controlled by a centralized authority, causing concerns regarding user privacy, bandwidth usage, and security of the user data. 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. Since decentralized and distributed web is not controlled by any third party, it is extremely beneficial in solving the above mentioned problems. This paper focuses on the components that form the basis of the decentralized web and investigates two technologies – Ethereum and the InterPlanetary File System (IPFS) to give the readers a brief introduction of the better, safer and faster web for everyone.**

Keywords—Decentralized Storage, Blockchain, Ethereum, InterPlanetary File System, P2P File Sharing

I. INTRODUCTION

In today's world, web and the internet by extension play a major role in the communication of digital information in all its forms. Web and its accompanying protocols such as HTTP, HTTPS etc. have played an instrumental role from the advent of the 1989 Web Revolution. We have had two major versions of Web till date i.e. Web 1.0 and Web 2.0 [1]. The first generation of web, aptly named Web 1.0 took the first step and enabled users to only share and view information and not interact with it dynamically. Disadvantages of Web 1.0 included fewer number of writers as compared to large number of reader, rendering the hosted services slow and thus, unusable. With the introduction of Web 2.0, users were able to communicate with one another and were able to interact with online content. This gave rise to various services such as blogs and wikis. Web 2.0 also had various disadvantages in the form of security vulnerabilities. This included Cross Site Scripting, SQL Injection, authentication and authorization flaws, etc. [2]. A decentralized web will be able to solve the problems faced by any service based on client server architecture. The next section reviews the components of the decentralized web, current technologies in the field, and the possible drawbacks one might face.

II. LITERATURE SURVEY

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 [3]. 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 [4, 5].

B. Distributed Hash Table

Distributed Hash Tables are used as a lookup service in distributed and decentralized services [4]. They are used to map identifiers from a common pool of peers or nodes in an overlay network [6]. 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 [4]. Most of the existing DHT assume that its peers are spread over the ID space uniformly [6].

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 [7, 8]. These ledgers are visible and using a P2P approach, the peers or nodes in the blockchain network can edit the distributed ledger [7, 9, 10,

11]. 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 [12, 13].

Working of a blockchain network [8, 14, 15, 16]:

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 [7]. The figure below depicts a blockchain [7].

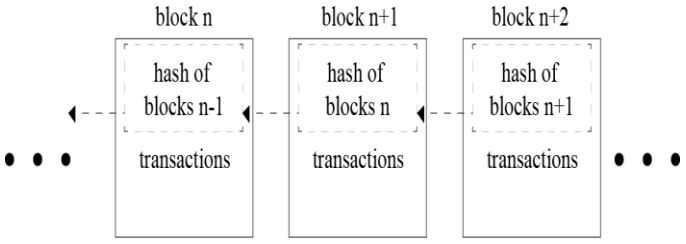

Fig. 1. Blockchain

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 [16]. IPFS uses the public key cryptography used in SFS [17].

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 [7, 10]. 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 [7, 10]. Some other types of Consensus Protocols are Delegated Proof of Stake (DPoS), Proof of Activity (PoA) - an amalgamation of PoW and PoS [7].

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 [14].

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” [18, 19]. 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 [14]. The high-level programming languages used for writing smart contracts are mainly Solidity, Serpent and Low-level Lisp-like Language (LLL) [18].

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 [19, 20]. Huge number of open source software systems are maintained by Git [20]. Version control system allows multiple users to store changes and switch and access different versions with ease [19].

Master branch is the current working branch which 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 [21].

Fig. 2. Forking and merging of Git branches

III. Existing Technologies

The blockchain is a relatively new approach in the field of information technology [12]. The complexity of the technology poses many challenges and foremost amongst these are management and monitoring of blockchain based decentralized applications [22]. 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 [12, 22]. The first live Ethereum blockchain network was launched in 2015 [22]. Ethereum cryptocurrency token "Ether" as of July 2018, third in popularity after Bitcoin and Ripple, is used to compensate participating peers for computations performed [14, 22].

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 [6, 23].

The consensus in the Ethereum network is based on modified GHOST protocol (Greedy Heaviest Observed Subtree) [24]. 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 [6].

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 [17]. IPFS, is a peer-to-peer distributed file system, aims to replace HTTP and build a better web for us all [25]. The torrent protocol facilitates relocation of data between nodes comprising the infrastructure and the Kademlia DHT is used for the management of metadata [7]. 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 [14, 17]. 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 [17]. Figure 3 depicts the creation of a new object: the client sends its object to any node on its nearest site [26].

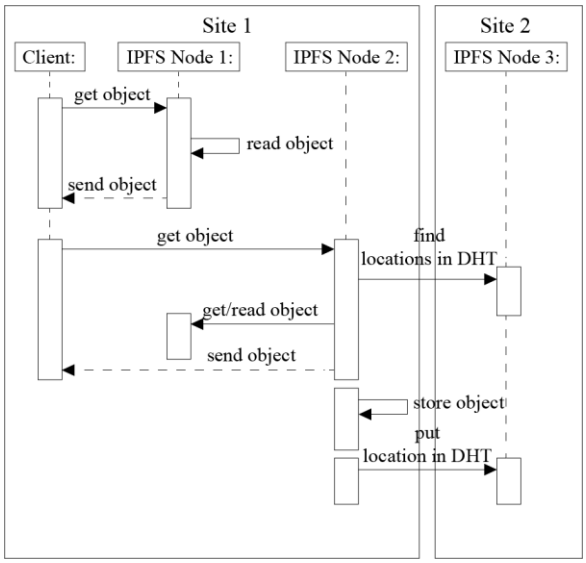

Fig. 3. Reading an IPFS Block [26]

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 [26]. Figure 4 illustrates what happens when the client reads an object stored on its local site [26].

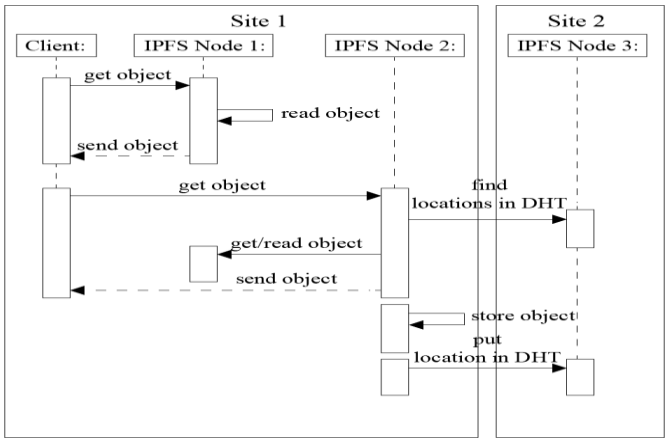

Fig. 4. Writing a block to IPFS [26].

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 [17, 26]. 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) [26].

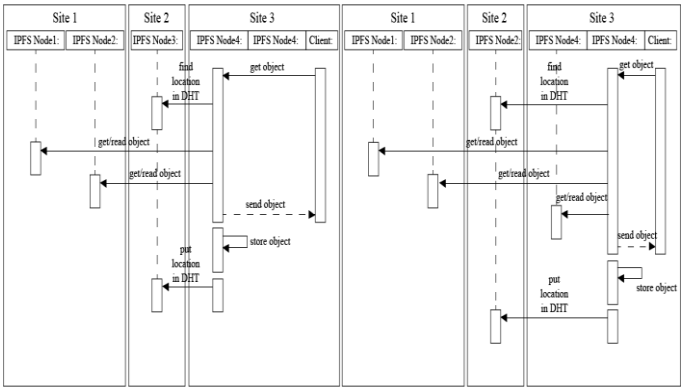

FIG. 5. Read an object stored remotely

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 [17, 26].

In the network model of decentralized services, applications distribute their workload over multiple hosts [27]. In some cases, the DApp is composed of components, each running on any random and independent host that provides the requisite services [28]. 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 [27]. 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 [29].

IV. CONCLUSION

The number of people using internet is increasing exponentially on a daily basis. Several upgrades have been made to the infrastructure supporting the current web, such as optic fiber cables, better servers etc. But such advances only postpone a bigger problem at hand. The current web is not secure, fast and robust. It is extremely vulnerable to Distributed Denial of Service attacks(DDoS) and hence, Web 2.0 will not remain relevant for long and it is necessary to move from Web 2.0 to a decentralized web. The client server architecture currently in use puts tremendous load on the servers to cater the ever increasing growth of the users. P2P architecture solves this by distributing the load among multiple peers who are currently using the service, eliminating the need of the centralized servers and providing faster, safer and private access to the content. IPFS is one such technology, which uses multiple peers in its network spread all across the world to transfer files without the need of a server. The restriction in data access through a decentralized application can be bypassed on the Ethereum blockchain and this fact can be utilized in the existing services. Applications of blockchain, once a far-fetched goal, are finally making inroads in the mainstream. The idea of distributed consensus is proving to be a prime substitute to maintain security. The introduction of smart contract has radically transformed traditional contracts. People are opting to use smart contract instead of a traditional contract as it is safe, tamper-proof and has a wide variety of functionality.

V. Future Work

The future work includes creating a Video Streaming platform as a Ethereum DApp (Decentralized App), which is an application that is deployed on the Ethereum Blockchain. The metadata would be saved on Ethereum Smart Contracts for easy accessibility. But since, we have to pay to use the Ethereum Blockchain, saving large files like the video files we aim to store on this would turn out to be cost inefficient. Thus, to save large files, using the IPFS would be economical. The Merkle DAG, the unique identifier for any file will be saved on the Ethereum Blockchain, along with other metadata such as title, author, size, quality etc. This will cause retrieval of the video from IFPS. This endeavor will thus make the video platform fast, secure and robust, making content accessible in areas with low bandwidth.

References

[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. 125-128, 2009.

[2] Keshab Nath, Sourish Dhar, Subhash Basishtha, "Web 1.0 to Web 3.0 - Evolution of the Web and its Various Challenges", IEEE, International Conference on Reliability, Optimization and Information Technology pp. 86 – 89, 2014.

[3] J. Blackburn, K. Christensen, "A Simulation Study of a New Green BitTorrent," IEEE, Proc. Green Communications Workshop in conjunction with IEEE ICC'09, 2009.

[4] Olaf Landsiedel, Stefan Gotz, Klaus Wehrle, "Towards Scalable Mobility in Distributed Hash Tables" IEEE, *Peer-to-Peer Computing*, 2006.

[5] Ling Zhong, Xiofan Wang, Maria Kihl, "Topological Model and Analysis of the P2P BitTorrent Protocol", IEEE, Proceedings of the 8th World Congress on Intelligent Control and Automation, pp. 754-758, 2011.

[6] 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.

[7] Lakshmi Siva Sankar, Sindhu M, M. Sethumadhavan, "Survey of Consensus Protocols on Blockchain Applications", IEEE, International Conference on Advanced Computing and Communication Systems, 2017.

[8] Donhee Han, Hongjin Kim, Juwook Jang, "Blockchain based Smart Door Lock System", IEEE, Information and Communication, pp. 1165, 2017.

[9] Jatinder Singh, John David Michels, "Blockchain as a Service (Baas): Providers and Trust", IEEE, European Symposium on Security and Privacy Workshops, pp. 1165, 2018.

[10] 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.

[11] 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.

[12] Dejan Vujičić, Dijana Jagodić, Siniša Ranđić , "Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview", IEEE, 17th International Symposium INFOTEH-JAHORINA, 2018.

[13] 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.

[14] Sreehari P , M Nandakishore, Goutham Krishna, "SMART WILL: Converting the Legal Testament into a Smart Contract", IEEE, International Conference on Networks & Advances in Computational Technologies, pp. 203-207, 2017.

[15] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", White Paper, 2008.

[16] David Mazieres, "Self Certifying File System", Doctor of Philosophy, Massachusetts Institute of Technology, Massachusetts, USA, 2000.

[17] Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System (Draft 3)", White Paper, 2014.

[18] 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.

[19] R. G. Shirey, K. M. Hopkinson, K. E. Stewart, "Analysis of implementations to secure git for use as an encrypted distributed version control system", IEEE, 48th Hawaii International Conference on System Sciences, pp. 5310–5319, 2015.

[20] 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.

[21] HaeJun Lee, Bon-Keun Seo, Euiseong Seo, "A Git Source Repository Analysis Tool Based on a Novel Branch-oriented Approach", IEEE, International Conference on Information Science and Applications (ICISA), 2013.

[22] Nida Khan , Abdelkader Lahmadi , Jerome Francois and Radu State, "Towards a Management Plane for Smart Contracts: Ethereum Case Study", IEEE, IFIP Network Operations and Management Symposium, 2018.

[23] V. Buterin, "Ethereum white paper: a next generation smart contract & decentralized application platform," Ethereum White Paper, 2013.

[24] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin,", Springer, Financial Cryptography, pp. 507-527, 2015.

[25] Yongle Chen, Hui Li , Kejiao Li and Jiyang Zhang, "An improved P2P File System Scheme based on IPFS and Blockchain", IEEE International Conference on Big Data, pp. 2652- 2657, 2017.

[26] Bastien Confais, Adrian Libre,Benoˆıt Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure based on IPFS and Scale-out NAS", IEEE, 1st International Conference on Fog and Edge Computing (ICFEC), 2017.

[27] Ludwig, Simone. "Comparison of Centralized and Decentralized Service Discovery in a Grid Environment." *Fifteenth IASTED International Conference on Parallel and Distributed Computing and Systems*, vol. 1, pp. 12- 17, 2003.

[28] Smith, Jerry. "*Distributed Computing with Aglets"*. White Paper, 1999.

[29] Robert W. Lucky, "The Lure of Decentraisation", IEEE Spectrum, pp. 23, 2017.