Secure Storage and Access for Task-Scheduling Schemes on Consortium Blockchain and Inter-planetary File System

Dongcheng Li
Department of Computer Science
University of Texas at Dallas
Richardson, USA
dxl170030@utdallas.edu

W. Eric Wong*
Department of Computer Science
University of Texas at Dallas
Richardson, USA
ewong@utdallas.edu

Man Zhao
China University of Geosciences
Wuhan, China
zhaoman@cug.edu.cn

Qiang Hou
China University of Geosciences
Wuhan, China
houqiang@cug.edu.cn

***Abstract*—Computerized systems and software, which allow optimizing and planning the processes of production, storage, transportation, sale, and distribution of goods, have emerged in the industry. Scheduling systems, in particular, are designed to control and optimize the manufacturing process. This tool can have a significant effect on the productivity of the industry because it reduces the time and cost through well-defined optimization algorithms. Recently, the applicability of blockchain technology has been demonstrated in scheduling systems to add decentralization, traceability, auditability, and verifiability of the immutable information that this technology provides. This is a novel contribution that provides scheduling systems with an additional layer of security. With the latest version of Hyperledger Fabric, the appropriate levels of permission and policies for access to information can be established with significant levels of privacy and security, which prevent malicious actors from trying to cheat or abuse the system. Different alternatives exist to manage all processes associated with the operation of a blockchain network, and among them, providers of blockchain as a service have emerged. Chainstack stands out for its simplicity and scalability features to deploy and operate a blockchain network. Our goal in this work is to create a solution for secure storage of and access to task-scheduling scheme on the consortium blockchain and inter-planetary file system as a proof of concept to demonstrate its potential and usability.**

Keywords—Blockchain, task scheduling, inter-planetary file system (IPFS), blockchain as a service (BaaS), Hyperledger Fabric, chaincode, DApps

I. INTRODUCTION

A significant effort has been made to optimize processes in scheduling systems. However, in terms of privacy and security, members must transfer their data and trust the security that system administrators implement in the system. In a traditional scheduling system, the tasks and execution times are generated by optimization algorithms. These tasks are stored in a centralized database where users consult, implement, and execute the tasks according to their role within the system [1]. The centralization of the information can be observed not only in the storage of the information but also in the validation of the algorithms and verification of the scheduled task execution.

As noted by Huang et al. [2], programming scheduling can be optimized by including the benefits of a blockchain network, such as the adequate handling of data. In general, the information is openly available to all authorized members of the network in a blockchain-type decentralized storage system, so that data are provided that promote additional confidence [3]. Blockchain technology can also provide a scheduling system with fault tolerance and an immutable history of the data due to blockchain's decentralized nature, cryptographic algorithms, and consensus methods. A malicious agent of a blockchain network must control a majority domain of the nodes to make modifications to the data stored in the blocks of the chain, which is infeasible for hackers and adversaries.

The consensus method can also be used to verify the validity of the system's scheduling algorithms and inner business logic. Before the information is added to the blocks, a data verifiability algorithm is run so that the system behaves in the way it was planned and not in a different way or with malicious intent. In this paper, we propose a solution for secure storage of and access to a task-scheduling scheme that uses a permissioned blockchain based on the Hyperledger Fabric v2 framework integrated with the inter-planetary file system (IPFS) service through Pinata and deploys on Chainstack with a web client based on Vue.js.

The preceding sections of the paper are arranged as follows. Section 2 reviews the related studies on blockchain technology and IPFS. Section 3 presents the Hyperledger Fabric blockchain technology and the motivation for using it on task scheduling. Section 4 introduces the blockchain as a service (BaaS), and the IPFS is explained in Section 5. Section 6 presents the proposed solution for the study. Finally, the paper is concluded in Section 7.

II. RELATED WORK

The health sector generates a massive amount of electronic data. The mismanagement of the data or the absence of tools that allow their secure storage is a fertile area for study. Sun et al. [4] devised an innovative encryption system to efficiently and securely store electronic medical

978-1-7281-8915-4/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS-C51114.2020.00035

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

data. They used an attribute-based encryption system combining blockchain technology and IPFS storage, which prevents access by intruders to vulnerable information and provides tamper-proof evidence in case of a medical dispute.

Document sharing and version management are other areas in which both blockchain and IPFS have provided tools for elaborate solutions or accurate improvements to the accomplishment of these tasks. Taking advantage of the benefits of these technologies, Nizamuddin et al. [5] proposed a solution that allows sharing documents and managing versions in a reliable, secure, and decentralized way, for which they used Ethereum's smart contracts and IPFS. The proposal they made automates the necessary interactions involving several participants, who are developers and quality assurance testers. The system is fully decentralized and was subjected to tests and demonstrated that the developed smart contract is resistant to commonly known attacks.

In the process of publishing scientific articles and peer reviews, the traditional method can be slow, centralized, and prone to unfairness; within this area, Tenorio-Fornés et al. [6] proposed an IPFS and blockchain-based decentralized publication system. In their work, they created a distributed system to determine the reputation of the reviewer and designed an open-access infrastructure, guaranteeing a transparent governance process. They also used a survey to assess problems and propose solutions and possible adoptions and created a working version that serves as a proof of concept.

As a result of the review of these studies, we observe that blockchain technology and distributed systems are much more than theory and are effectively being well-implemented, tested, and used. They offer concrete solutions at the business level. Based on this experience, we propose using these technologies along with task scheduling to design a solution for secure storage of and access to task-scheduling schemes on blockchain and IPFS.

III. HYPERLEDGER FABRIC BLOCKCHAIN TECHNOLOGY

Hyperledger Fabric is an open-source platform for the consortium blockchain started by the Linux Foundation. The objective is to support enterprise-grade blockchain applications beyond cryptocurrencies. In Hyperledger Fabric, transactions are conducted by users with known identities and defined permissions and access to data. It offers its participating organizations the ability to choose the desired consensus mechanism and membership service providers. Unlike most public blockchains, a new block of the Fabric network can be added without the possibility of future forking if a significant fraction of all validators confirm the content within the block. Moreover, the Fabric network is a private network where private transactions within a group of selected participants can be achieved. It also provides a high level of security and auditability due to the design of the system. Altering the information in the blockchain is computationally infeasible.

A. Evolution of Different Versions of Hyperledger Fabric

Since the first version of Hyperledger Fabric released in mid-2017, several updates have been made that have introduced changes in the security level. Issues reported by the community have been resolved, and vulnerabilities have been overcome in a process of continuous improvement. In addition, larger changes have been made that involve a) important modifications at the architecture level, b) the scalability, and c) the security of this blockchain. We evaluate the main versions of Hyperledger Fabric from version 1.0 to the current version (2.1), which is the one we used in this work.

Version 1.0 of Hyperledger Fabric adapted the new *execute-order-validate* as its default architecture and added the option of CouchDB to its World State database [7]. *Execute-order-validate* ensures the finality of the system, and the CouchDB increases the performance of the state database in some circumstances. However, this version of Hyperledger Fabric does not support any Byzantine fault-tolerant based ordering service, which leads to a limitation in its system's fault-tolerance capability [8]. There also exist the risk of a nondeterministic transaction occurs due to a read-write conflict. The functionality of peer nodes, data maintenance, and consensus services of the blockchain were separated in later versions of Fabric [9]. Unlike the previous versions of Fabric, version 1.1 takes advantage of vCPUs and achieves parallel processing for committing peers [10]. Starting from version 1.2 of Hyperledger Fabric, the network can be integrated with the Kafka consensus algorithm, which is based on permissioned voting and offers production-level functionalities. Although this method provides crash fault tolerance, it also causes intimidating administrative overhead. Hyperledger Fabric version 1.3 introduced a high-level application programming interface (API) to conduct smart contract implementations and key-level endorsement policies that can simulate different endorsement policies for different variables [11].

The progress made in Hyperledger Fabric 1.4 includes several mechanisms for obtaining system performance metrics. In this sense, the level of detail of the system logs is also increased so that the administrator can be aware of the state of the network. At the user-interface level, personalized notifications are sent regarding the expiration of the certificate, and known issues are solved at the chaincode and login level in the web client. Authors who have used Hyperledger Fabric for the elaboration of their projects along with the version numbers are shown in Table 1.

TABLE I. HYPERLEDGER FABRIC VERSIONS USED IN RECENT RESEARCH

Fabric v1.0.x	**Fabric v1.1.x**	**Fabric v1.2.x**	**Fabric v1.3.x**	**Fabric v1.4.x**
[10], [12], [13], [14]	[15], [8], [16]	[17], [7], [18], [19], [20]	[11], [21]	[22]

B. Hyperledger Fabric v2

Significant and novel changes have been introduced since beta version 2.0, emphasizing greater decentralization and modularity with important changes at the level of

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

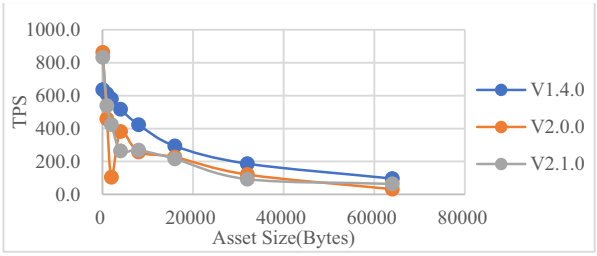
data security and privacy [23]. Starting with this version, a new decentralized governance scheme is available through the use of smart contracts. This is accomplished through a novel life cycle of the chaincode, which makes it possible to reach an agreement on the parameters used by the chaincode prior to interaction with the blockchain [24].

The developers put a significant extra effort on privacy for this new version. From now on, the way to share private data can be done singularly by an organization or directly to the regulator or auditor of each organization and not necessarily to the entire collection of members. Similarly, the way to verify the authenticity of private data is through the chaincode API, comparing the hash of the information provided by the user. The raft-based consensus is now natively supported by Hyperledger Fabric v2.0, and the Kafka-based consensus implemented in the previous versions of Fabric was deprecated in Hyperledger Fabric v2.0. However, since version 2.0, the user can migrate from the Kafka consensus to the raft consensus method [25].

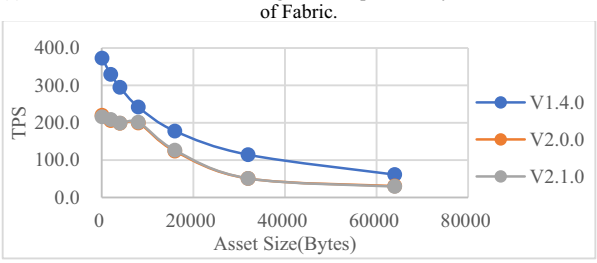
C. Performance Evaluation on Most Recent Versions of Hyperledger Fabric

We compared the performance of the three major releases of Hyperledger Fabric (v1.4, v2.0, and v2.1). The original data for each version of Fabric came from [26], which collected by running Hyperledger Caliper, a performance benchmarking tool, against the blockchain network. Even though the Hyperledger Fabric supports both LevelDB and CouchDB, the comparison only considers the performance with LevelDB because Chainstack, the platform used in our implementation, currently only supports LevelDB in the production environment. We recognize that deviations may be caused by subtle differences in the Caliper versions used to test different versions of Fabric and by the slight differences in hardware when testing versions 1.4 and 2.x. Nevertheless, the graph still grants some insight into this study. Figure 1 reveals that Fabric version 1.4 is more stable than Fabric version 2.0 and 2.1 in terms of throughput (TPS) performance under different asset sizes. However, versions 2.0 and 2.1 outperform 1.4, exhibiting superior maximum throughput speed when extracting and returning a single asset from the World State database, as illustrated in Figure 1(a). However, as for the performance of inserting an asset into the World State database, version 1.4 exhibits greater overall throughput than versions 2.0 and 2.1, as displayed in Figure 1(b).

Compared to the previous versions of Hyperledger Fabric, version 2.0 has introduced higher levels of security and protection for user data [23]. This is why this version seems adequate for the development of a task-scheduling system. It is of interest to a network that uses sensitive data to maintain the security of the user data and the tasks that are planned. Furthermore, a blockchain-based system applied to this environment guarantees that the tasks and planned schemes are not modified by the personal interests of the users, thus promoting confidence in the rest of the network that the information processed is in the collective interest and adheres to the planning agreed by consensus. At the time of writing this paper, Hyperledger Fabric version 2.0 and 2.1 has not been used in any research paper as far as we investigated.



(a) Performance evaluation of the get-asset operation by different versions of Fabric.



(b) Performance evaluation of the create-asset operation by different versions of Fabric.

Figure 1. Effect of the size of assets and the release version of Fabric on the throughput of transactions using LevelDB.

IV. BLOCKCHAIN AS A SERVICE

Implementing a blockchain from scratch involves managing a series of elements that are not intuitive at first glance. The nodes that interact with each other, must be synchronized at the network level, handle the validations of consensus methods, the web client with which the user interacts with the system, and more. If a system administrator aims to control all of this, he or she can be overwhelmed by the complexity of services that interact to keep the network working. In this sense, BaaS providers have emerged, which allow the business to use predefined solutions to create, build, and host the necessary components in the cloud for the operation of a blockchain [27]. The service administrator contracts the services of the provider, which allows the administrator to take advantage of the necessary resources, such as managing the identity of the users, adding privacy elements, and escalating user privileges, along with security for the storage of information in the cloud.

Moreover, these BaaSs provide a set of tools to reduce the development time of distributed applications and facilitate the tasks of preparing the blockchain environment so that the application can run. For the administrator of a blockchain, it is also important to know in real time the performance, state of the network, and costs associated with the services purchased and to manage the storage and processing capacity. Therefore, BaaS also includes different tools for monitoring and controlling the system with early alerts to notify the administrator of the system status. An example is depicted in Figure 2. In the following sections, we evaluate various providers and present a succinct comparison of their benefits and why we chose Chainstack to simulate the use case.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

A. Blockchain as a Service Supports Hyperledger Fabric

Among the available services that support Hyperledger Fabric we have mainly IBM Blockchain Platform, Amazon Managed Blockchain, and Chainstack Managed Blockchain. The IBM Blockchain Platform deserves special mention since it worked with the Linux Foundation to create the first version of Hyperledger Fabric [28]. In this sense, it is a platform that offers greater maturity, more complete documentation, and greater use-case exploration [12]. The IBM Blockchain Platform also offers its users the choice of the deployment environment, such as multi-cloud, hybrid, or on-premises deployment. Moreover, it currently supports Hyperledger Fabric up to version 1.4 [29].

In contrast, Amazon Managed Blockchain offers the integration of the network with Amazon's services, Amazon Web Services [30], to obtain an operational application. Businesses that generally choose Amazon Managed Blockchain to develop and deploy their blockchain application do so because they have already used other services from AWS, which can integrate the data stored in the ledger and extract the analyses. This service also promise to offer a template for the Ethereum public blockchain platform in the future, not just for the Fabric platform, and supports the current versions of Fabric up to 1.2 [12]. In addition, with AWS, a free tier may be used with limited capacity for the development process. Moreover, Cloud9, a cloud-based integrated development environment, allows developers to write, run, and debug code for blockchain applications with only a browser. The user has access to the monitoring tool at all times to control the cost of storage and processing [30].

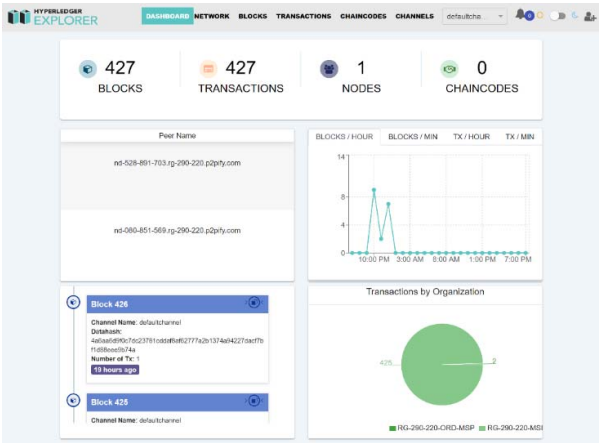


Figure 2. Hyperledger Explorer integration with the Chainstack-managed Hyperledger Fabric v2 blockchain.

B. Chainstack

We highlight Chainstack, a newly emerged BaaS, as our choice to display the product of this work. After an exhaustive evaluation of the previous options, we observed that Chainstack has a series of benefits that are better adapted to our use case. In the first instance, Chainstack supports the latest version of Hyperledger Fabric v2.1. The current features of the membership service provider component are in line with the latest version of Hyperledger Fabric in terms of privacy and handling user data [31]. According to the experience and collection in the research that is referenced in this work, Chainstack has a more intuitive service and greater flexibility when orchestrating the services associated with Hyperledger Fabric [32]. Table 2 compares the BaaSs that support the Hyperledger Fabric protocol. In addition, Chainstack has done tremendous work to support consortium blockchain projects; thus, its platform offers greater advantages that are adapted to our particular case given that our interest is focused on the benefits provided by version 2.x of Hyperledger Fabric.

TABLE II. FEATURE COMPARISON FOR BLOCKCHAIN AS A SERVICE (BAASS) SUPPORTING THE HYPERLEDGER FABRIC PROTOCOL[30],[31],[33]

	Amazon Managed Blockchain	IBM Blockchain	Chainstack
Currently support Hyperledger Fabric version	1.2	1.4	2.1
Provide choices of framework (Public or Consortium)	✗	✗	✓
Provide choices of deploy environment (Hybrid or Cloud)	✗	✓	✓
Integration with existing cloud services	✓	✓	✗
Provide provisioning automation	✗	✗	✓
Provide network configuration Automation	✗	✗	✓
Has compatible VS Code blockchain extension	✗	✓	✓
Can develop using just a browser/online IDE	✓	✗	✗

V. INTER-PLANETARY FILE SYSTEM

The IPFS is a decentralized, verifiable, blockchain-compatible storage system. Most business solution application areas that use the blockchain may require the use of attachments to support transactions, for example, invoices, property contracts, delivery notes, images, videos, or physical endorsements of certificates. In these cases, the interoperability between the IPFS protocol and Hyperledger Fabric is used to store a hash with which the digitized document is associated, whether it is an image, PDF, txt, Word, or any other type of file. This is necessary because, within a blockchain, storing data is expensive. Therefore, the IPFS service allows what is stored in the transaction within a blockchain to be just a hash that points to the file stored in the IPFS. The cryptographic method that IPFS uses is compatible with Hyperledger Fabric and ensures that the file remains unchanged because it is associated with a unique digital mark [34].

A. Basics of Inter-planetary File System (IPFS) and IPFS Gateway

Unlike the way the Internet works by addressing the content location through a URL, IPFS works with a protocol that allows direct access to content using a content identifier based on a cryptographic hash, which guarantees the inalterability of the data stored. In addition, IPFS works

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

in a decentralized manner. When a document or file is uploaded to IPFS, each node on this network can access this content. This allows the content search to be faster and access to the content to be more resilient. Being shared in different nodes that set up the IPFS network, it does not rely on a single server or database [35], which eliminates the risk of a single point of failure. Moreover, services called IPFS gateways allow web users to upload and retrieve data on the IPFS network without installing and running the IPFS node on their machines. The IPFS gateways also allow developers of the blockchain applications to use the IPFS services in the backend with ease. Unlike other services, our preferred IPFS gateway service Pinata allows store diverse types of data, which increases the potential use cases of our blockchain solution.

B. Inter-planetary File System and Blockchain

In the blockchain, all information about the transactions that occur in the network must be stored in all nodes and must be validated using the consensus method; thus, it is not computationally practical to transmit and store large amounts of information and is often avoided by the developers and designers of blockchain systems, which significantly limits blockchain capabilities. In this context, when we store information in IPFS, we call this storage of information *off-chain*, and the information that is directly stored in the blockchain is *on-chain*. If we know the hash of content in IPFS, we can access the content; however, it is difficult to link the content to a person or entity.

Based on this principle, Kumar et al. [36] designed off-chain distributed storage for medical patient data where the hash of the documents is stored on a consortium blockchain, thus preserving the integrity of the data and privacy of the patients. The protection of information by maintaining the integrity of the data was studied by Agyekum et al. [37], where the IPFS and blockchain were combined to produce an architecture focused on protecting the integrity of copyright and data security. Hoffman et al. [38] defined a decentralized application that includes a reliable payment system, a decentralized reporting method, and a guarantee of the immutability of the data transmitted by storing the bug reports in IPFS and handling the rest of the transactions in the Ethereum blockchain. The integration of IPFS can dramatically reduce the asset size of a single transaction on-chain because it only stores the hash of the content. Moreover, the smaller the asset size, the better the performance is in terms of the throughput of a Hyperledger Fabric blockchain, as illustrated in Figure 2.

Finally, taking into account the successful experiences of the studied architectures, we decided to integrate the decentralized off-chain storage IPFS, in particular through Pinata, together with the Hyperledger Fabric v2 Blockchain to produce a robust system in terms of security, integrity and the availability of data for a satellite task scheduling and planning scheme management system.

VI. EXPERIMENT SIMULATION

In this section, we present a blockchain-based permissioned and distributed satellite task-scheduling scheme management system that uses a consortium blockchain, based on the Hyperledger Fabric v2 framework with raft consensus. This development uses the BaaS provided by Chainstack to host blockchain peers and expedite the network formation. It also integrates with the IPFS service through Pinata, an IPFS gateway, to handle off-chain storage and Vue.js for the web client. The main objective of this experiment is to provide security and tamper resistance to the task-scheduling scheme of satellite observation and verify the feasibility of combining the Hyperledger Fabric v2 blockchain framework with IPFS. The flow of the system is presented in Figure 3.

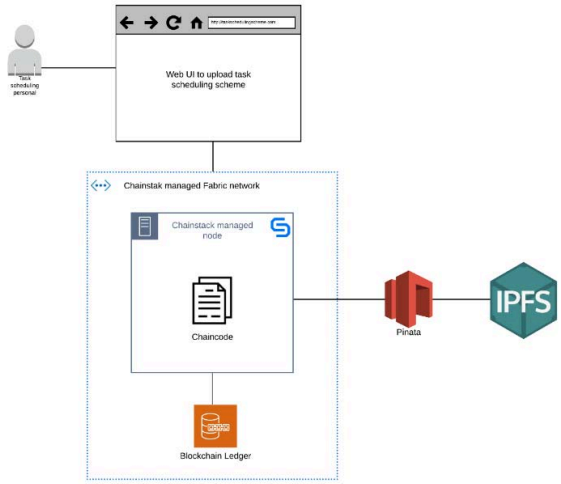
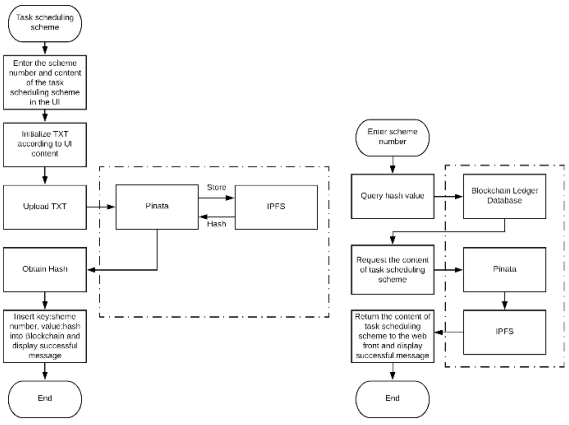


Figure 3. System flow.



(a) Insert operation. (b) Query operation.

Figure 4. The application process for an insert and query operation using the blockchain application.

A. Setup

A typical peer node on Chainstack has 1.3 vCPU and 2 GB of RAM initially allocated that can be either running on the cloud or in a hybrid environment. In our case, we selected the Google Cloud platform in Singapore to host our single raft-based orderer and peer nodes. Resource allocation is elastic; thus, the nodes can be scaled dynamically. For the Fabric protocol, official Docker images that run Alpine Linux were used. Currently in production, Chainstack supports LevelDB out of the box. Meanwhile,

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

CouchDB is in the testing phase and will ready to be released later. In addition, the node.js application we used to communicate with the Fabric blockchain network was developed on an Ubuntu 16.04 LTS operating system with 4 GB of RAM. The detailed development environment prerequisites are listed in Table 3.

TABLE III. DEVELOPMENT ENVIRONMENT PREREQUISITE

Software Prerequisites	Version
Ubuntu Linux	16.04 LTS
Docker	18.09.1
Docker Compose	1.24.1
Node.js	8.15.0
axios	0.19.2
npm	6.4.1
Pinata	1.1.8
Fabric	2.1.0

B. Flow of the Application

The scheduling algorithms are used to generate outputs based on various inputs in the planning system. The output is a set of predetermined task-scheduling/planning schemes for satellite observation. This set of tasks are stored in the decentralized storage IPFS, where a unique hash is generated for the scheme. Then, a transaction in the blockchain is generated, and this hash and its associated task-planning scheme number are written into a block. This is accomplished through a blockchain application and chaincode without the need to host an IPFS full node. The next step is adjusting the input of the scheduling algorithms to produce different task-planning schemes until a final planning scheme is determined. This process is performed by the designated personnel. Once the planning scheme is written in a transaction on-chain, it guarantees the immutable existence of such a planned scheme, and every changes/modification happens to this scheme beyond this point. The benefit of this is preventing a malicious user with a particular interest from manipulating the system for their benefit. Figure 4 reveals the steps for our decentralized application (DApps) to perform a typical insert or query operation.

VII. CONCLUSION

Blockchain technology has evolved in recent years and is increasingly applied to a broader range of businesses and systems where traditional technology required an additional boost in the areas of data privacy, security, availability, and confidentiality. Hyperledger Fabric, in its latest version, has had a significant improvement in terms of security and privacy of user data. On the other hand, the combination of the blockchain application and IPFS technology adds an additional layer of decentralization, availability, anonymity, and immutability to the data within and allows storing diverse types of data off-chain.

In this paper, we developed a decentralized, and permissible blockchain-based application for storing and accessing satellite task-scheduling schemes as a proof of concept, which is supported by the Hyperledger Fabric v2 framework and IPFS off-chain storage. This system has been deployed in Chainstack, and the web interface was created using Vue.js. To further study the benefits of BaaS, the features of the popular BaaS that support Hyperledger Fabric were explored and compared. The results indicate that Chainstack has a series of benefits that are better adapted to our use case, such as support for the latest version of Hyperledger Fabric, intuitive and easy network formation and node setup, and the capability of producing a stable and production-ready blockchain application, which are vital to our implementation.

Moreover, IPFS was introduced into the consortium blockchain application to offload on-chain storage, which reduced the asset size of the transaction and increased the transaction throughputs of the network. Additionally, the performance of the major versions of Fabric under different asset sizes was analyzed and compared, and the results indicate that the smaller the asset size, the greater the performance of the throughput. and by integrating IPFS with Hyperledger Fabric, the size of the asset in a single transaction can be controlled and minimized.

At the time of writing this paper, the latest version of Hyperledger Caliper is v0.3.0, which does not fully support Hyperledger Fabric v2.1.1 [39]. In future work, Hyperledger Caliper will be used to test and assess the performance of our blockchain-based system when Hyperledger Fabric v2 support is fully implemented in Hyperledger Caliper.

REFERENCES

[1] J. M. Framinan, R. Leisten, and R. R. García, "Overview of Scheduling Systems," *Manufacturing Scheduling Systems*, pp. 337–352, 2014.

[2] X. Huang, Y. Zhang, D. Li, and L. Han, "An optimal scheduling algorithm for hybrid EV charging scenario using consortium blockchains," *Future Generation Computer Systems*, vol. 91, pp. 555–562, 2019.

[3] G. Zyskind, O. Nathan, and A. 'sandy' Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," *2015 IEEE Security and Privacy Workshops*, 2015.

[4] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-Based Secure Storage and Access Scheme for Electronic Medical Records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[5] N. Nizamuddin, K. Salah, M. A. Azad, J. Arshad, and M. Rehman, "Decentralized document version control using ethereum blockchain and IPFS," *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[6] A. Tenorio-Fornés, V. Jacynycz, D. Llop-Vila, A. Sánchez-Ruiz, and S. Hassan, "Towards a Decentralized Process for Scientific Publication and Peer Review using Blockchain and IPFS," *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[7] S. Mazumdar and S. Ruj, "Design of Anonymous Endorsement System in Hyperledger Fabric," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2019.

[8] J. Sousa, A. Bessani, and M. Vukolic, "A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform," *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018.

[9] S. Zhang, E. Zhou, B. Pi, J. Sun, K. Yamashita, and Y. Nomura, "A Solution for the Risk of Non-deterministic Transactions in Hyperledger Fabric," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019.

[10] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee, "Performance Characterization of Hyperledger Fabric," *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2018.

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.

[11] L. V. Hoye, P.-J. Maenhaut, T. Wauters, B. Volckaert, and F. D. Turck, "Logging mechanism for cross-organizational collaborations using Hyperledger Fabric," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019.

[12] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, "Trusted Computing Meets Blockchain: Rollback Attacks and a Solution for Hyperledger Fabric," *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, 2019.

[13] N. Klaokliang, P. Teawtim, P. Aimtongkham, C. So-In, and A. Niruntasukrat, "A Novel IoT Authorization Architecture on Hyperledger Fabric With Optimal Consensus Using Genetic Algorithm," *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, 2018.

[14] Q. Nasir, I. A. Qasse, M. A. Talib, and A. B. Nassif, "Performance Analysis of Hyperledger Fabric Platforms," *Security and Communication Networks*, vol. 2018, pp. 1–14, 2018.

[15] A. Goranovic, M. Meisel, S. Wilker, and T. Sauter, "Hyperledger Fabric Smart Grid Communication Testbed on Raspberry PI ARM Architecture," *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2019.

[16] H. Javaid, C. Hu, and G. Brebner, "Optimizing Validation Phase of Hyperledger Fabric," *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2019.

[17] A. Sharma, F. M. Schuhknecht, D. Agrawal, and J. Dittrich, "Blurring the Lines between Blockchains and Database Systems," *Proceedings of the 2019 International Conference on Management of Data - SIGMOD '19*, 2019.

[18] Y. Manevich, A. Barger, and Y. Tock, "Endorsement in Hyperledger Fabric via service discovery," *IBM Journal of Research and Development*, vol. 63, no. 2/3, 2019.

[19] C. Schaefer and C. Edman, "Transparent Logging with Hyperledger Fabric," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019.

[20] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019.

[21] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on Hyperledger Fabric with secure multiparty computation," *IBM Journal of Research and Development*, vol. 63, no. 2/3, 2019.

[22] P. Cui, J. Dixon, U. Guin, and D. Dimase, "A Blockchain-Based Framework for Supply Chain Provenance," *IEEE Access*, vol. 7, pp. 157113–157125, 2019.

[23] "What's new in Hyperledger Fabric v2.0," *Hyperledger-Fabric*, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.0/whatsnew.html. [Accessed: 16-Jun-2020].

[24] "The Ordering Service," *Hyperledger-Fabric*, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.0/orderer/ordering_service.html. [Accessed: 17-Jun-2020].

[25] "Migrating from Kafka to Raft," *Hyperledger-Fabric*, 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.0/kafka_raft_migration.html. [Accessed: 17-Jun-2020].

[26] "Hyperledger Blockchain Performance," *Hyperledger Blockchain Performance Reports*, 2020. [Online]. Available: https://hyperledger.github.io/caliper-benchmarks. [Accessed: 17-Jun-2020].

[27] Q. H. Mahmoud, M. Lescisin, and M. Altaei, "Research challenges and opportunities in blockchain and cryptocurrencies," *Internet Technology Letters*, vol. 2, no. 2, 2019.

[28] "IBM Blockchain Platform for IBM Cloud," *IBM*, 18-Jun-2019. [Online]. Available: https://cloud.ibm.com/docs/blockchain?topic=blockchain-hyperledger-fabric. [Accessed: 17-Jun-2020].

[29] "IBM Cloud," *IBM*, 2018. [Online]. Available: https://cloud.ibm.com/doc. [Accessed: 17-Jun-2020].

[30] "Build and deploy an application for Hyperledger Fabric on Amazon Managed Blockchain," *Amazon Web Services*, 14-Jan-2019. [Online]. Available: https://aws.amazon.com/es/blogs/database/build-and-deploy-an-application-for-hyperledger-fabric-on-amazon-managed-blockchain/. [Accessed: 17-Jun-2020].

[31] "Fabric - Chainstack," *Chainstack*, 2020. [Online]. Available: https://chainstack.com/fabric/. [Accessed: 17-Jun-2020].

[32] "Deploy and manage high-performing Hyperledger Fabric v2 nodes and networks in minutes," *Chainstack*, 2020. [Online]. Available: https://chainstack.com/build-better-with-fabric/. [Accessed: 17-Jun-2020].

[33] "IBM Blockchain Platform," *IBM Cloud*. 2019. [Online]. Available: https://www.ibm.com/cloud/blockchain-platform. [Accessed: 17-Jun-2020].

[34] "What is IPFS?" IPFS. Docs Beta. [Online]. Available: https://docs-beta.ipfs.io/concepts/what-is-ipfs/#decentralization. [Accessed: 17-Jun-2020].

[35] "Content addressing and CIDs" IPFS. [Online]. Available: https://docs-beta.ipfs.io/concepts/content-addressing/. [Accessed: 17-Jun-2020].

[36] R. Kumar, N. Marchang, and R. Tripathi, "Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain," *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2020.

[37] K. O.-B. O. Agyekum, Q. Xia, Y. Liu, H. Pu, C. N. A. Cobblah, G. A. Kusi, H. Yang, and J. Gao, "Digital Media Copyright and Content Protection Using IPFS and Blockchain," *Lecture Notes in Computer Science Image and Graphics*, pp. 266–277, 2019.

[38] A. Hoffman, E. Becerril-Blas, K. Moreno, and Y. Kim, "Decentralized Security Bounty Management on Blockchain and IPFS," *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020.

[39] "Fabric Configuration," *Hyperledger Caliper*, 2020. [Online]. Available: https://hyperledger.github.io/caliper/v0.3.1/fabric-config/. [Accessed: 20-Jun-2020].

Authorized licensed use limited to: Univ of Texas at Dallas. Downloaded on October 10,2021 at 11:36:42 UTC from IEEE Xplore. Restrictions apply.