electronics

MDPI

*Article*

# Blockchain-Enabled Decentralized Secure Big Data of Remote Sensing

**Abdul Razzaq [1], Syed Agha Hassnain Mohsan [2], Shahbaz Ahmed Khan Ghayyur [3], Mohammed H. Alsharif [4], Hend Khalid Alkahtani [5,*], Faten Khalid Karim [6] and Samih M. Mostafa [7]**

[1] Ocean Technology and Engineering, Ocean College, Zhejiang University, Zhoushan 316000, China
[2] Optical Communications Laboratory, Ocean College, Zhejiang University, Zhoushan 316021, China
[3] Department of Computer Science and Software Engineering, International Islamic University, Islamabad 44000, Pakistan
[4] Department of Electrical Engineering, College of Electronics and Information Engineering, Sejong University, Seoul 05006, Korea
[5] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[6] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[7] Faculty of Computers and Information, South Valley University, Qena 83523, Egypt
* Correspondence: hkalqahtani@pnu.edu.sa

**Abstract:** Blockchain technology has emerged as a promising candidate for space exploration and sustainable energy systems. This transformative technology offers secure and decentralized strategies to process and manipulate space resources. Remote sensing provides viable potential with the coexistence of open data from various sources, such as short-range sensors on unmanned aerial vehicles (UAVs) or Internet-of-Things (IoT) tags and far-range sensors incorporated on satellites. Open data resources have most recently emerged as attractive connecting parties where owners have shown consent to share data. However, most data owners are anonymous and untrustworthy, which makes shared data likely insecure and unreliable. At present, there are several tools that distribute open data, serving as an intermediate party to link users with data owners. However, these platforms are operated by central authorities who develop guidelines for data ownership, integrity, and access, consequently restricting both users and data owners. Therefore, the need and feasibility of a decentralized system arise for data sharing and retrieving without involving these intermediate limiting parties. This study proposes a blockchain-based system without any central authority to share and retrieve data. Our proposed system features (i) data sharing, (ii) maintaining the historical data, and (iii) retrieving and evaluation of data along with enhanced security. We have also discussed the use of blockchain algorithms based on smart contracts to track space transactions and communications in a secure, verifiable, and transparent manner. We tested the suggested framework in the Windows environment by writing smart contracts prototype on an Ethereum TESTNET blockchain. The results of the study showed that the suggested strategy is efficient, practicable, and free of common security attacks and vulnerabilities.

**Keywords:** blockchain; smart contract; IPFS; remote sensing; space industry; big data; data sharing

check for updates

**Citation:** Razzaq, A.; Mohsan, S.A.H.; Ghayyur, S.A.K.; Alsharif, M.H.; Alkahtani, H.K.; Karim, F.K.; Mostafa, S.M. Blockchain-Enabled Decentralized Secure Big Data of Remote Sensing. *Electronics* **2022**, *11*, 3164. https://doi.org/10.3390/electronics11193164

Academic Editor: Habib Hamam

Received: 29 August 2022
Accepted: 26 September 2022
Published: 1 October 2022

## 1. Introduction

In the last few years, we have witnessed remarkable breakthrough innovations from academic and industrial experts to introduce fifth-generation (5G). Meanwhile, several initiatives have been launched to forecast the roadmap for future sixth-generation (6G) technology. 6G is envisaged to offer higher reliability, lower latency, higher data rates, massive connectivity, and high coverage extension as compared to 5G technology. 6G is envisioned to revolutionize several research domains such as remote sensing, big data management, unmanned aerial vehicle (UAV) communication, smart agriculture, smart healthcare, etc. Among these, securing big data of remote sensing is of paramount significance. One of the most essential features of document remote sensing data is collaborative cooperation, which increases credibility among the parties involved. One of the key problems in document remote sensing data is managing correct digital data and tracing alterations in the digital contents when numerous entities are engaged in the document's preparation. In today's fast-paced world, document remote sensing data is frequently utilized to speed up product development and release cycles. The shift to digitization has resulted in content inconsistencies and document collaboration difficulties, with remote sensing data concerns accounting for 83 percent of productivity [1]. In existing documents, remote sensing data solutions [2] are primarily centralized and have some shortfalls such as high time utility, inappropriate document updating methods, and document changes without the consent of network users. Furthermore, with centralized systems, the document's alterations and updated historical data might be intruded upon, compromising the authenticity of the alterations and their updated record. As a result, a fully secure and decentralized infrastructure for maintaining digital document versions is needed [3].

Blockchain has emerged as a transformative technology after the success of Bitcoin. Bitcoin's underlying technology is named blockchain technology. The lack of resource constraints, hardware, and software impairments, and the immature standardizations in the existing technologies are the main reasons to utilize blockchain technology [4]. Blockchain is a decentralized and distributed technology that can reliably overcome security challenges in internet-of-things (IoT) networks. Most of the existing authentication methods and security mechanisms are centralized and usually require a reliable third party [5], which can increase the cost and energy consumption due to additional overheads. On the other hand, blockchain is cost-efficient and can securely manage data without involving any third party. Blockchain entities can execute and validate data before the inclusion of blockchain data. This procedure is referred to as a consensus mechanism that entirely removes the participation of any third party during data processing. This mechanism ensures functional resilience, immutability, and data transparency of the blockchain ledger as well as avoids fraudulent activities. It can also leverage the public key framework for encryption in peer-to-peer networks, authorization of network entities, and data authentication. It supports decentralized storage of heterogeneous IoT objects, sharing between various entities, and secure data processing. However, the existing blockchain has some drawbacks such as it is not computationally smart enough to manage big data [6,7]. Multiple nodes contain a replica of the entire blockchain ledger. This feature demands high storage capacity. Consequently, the consensus mechanism cannot process speedy transactions like classical centralized approaches.

Based on the consensus process, blockchain creates a database or distributed ledger which is distributed among entire network members. The system is secure and totally decentralized without the use of third-party verification. Miner nodes, that retain a replica of the Blockchain ledger, digitally sign, verify, and validate each transaction which brings an update to the ledger. This results in tamper-proof ledgers that are totally decentralized, secure, time-stamped, and shareable [8]. Many industries have used blockchain technology, including banking, accounting, document management, logistics, supply chain, and healthcare [9,10]. Blockchain technology is used to address challenges like data sharing, privacy, security, efficiency, and trust because of its strong and decentralized infrastructure [11]. By utilizing the power of cryptography to create trustworthy strategies for the participants in the chain, this method mitigates the need for a third-party transfer entity.

Smart contracts are computer algorithms that may be executed through a network of mutually distrusting nodes, without requiring any trustworthy authority [12]. They have several advantages due to their resilience to tamper. Smart Contracts are software programs that may be performed using Blockchain nodes. A smart contract is referred to as a self-executing algorithm that can validate that preset terms and conditions are being followed [13]. Smart contracts can change blockchains into distributed computing architectures. In a recent study [14], authors have analyzed whether smart contracts can be deployed by a state-of-the-art blockchain to operate as a component technique to the computation paradigm in order to enhance cost-effectiveness and foster reuse. A blockchain mining node, rather than verifying digital currency like Bitcoin, executes, validates, and stores data in blocks. Consigning a transfer to its Ethereum contact and executing it based on the input supplied for that transfer is how a smart contract is activated. Ethereum is a blockchain-empowered, open-source, distributed network with smart contract capabilities, as defined in [15]. Users can build their own code on top of the Ethereum network, allowing for the creation of custom apps. Ethereum makes use of Ether in the form of a cryptocurrency to make payments for transactions on the Ethereum blockchain. An Ethereum Address is used to identify each member in the Ethereum network (EA).

The blockchain is inefficient when it comes to storing an excessive amount of data. The major hurdle in the use of blockchain technology is its poor performance. It has, however, been shown to be successful when document hashes in the chain are stored rather than the documents themselves. When any data is uploaded to IPFS, a hash is created, which is then saved in the smart contract that is used to retrieve the content. For any modifications made to the document's content, the hash value changes every time. Existing distributed remote sensing data systems are typically centralized, which means they are generally controlled by a single central database, and users cannot get full control over the file or document [16]. Documents can be destroyed, altered, or tampered with using centralized systems. Furthermore, a developer or user connected with their account possesses the ability to edit items saved on the main server, based on current distributed remote sensing data systems.

Each remote sensing dataset in Interplanetary File System (IPFS), IPFS is a distributed and peer-to-peer content-based protocol that integrates smart contracts (SM) with blockchains data, is given a cryptographic hash to encrypt the text and make it unchangeable [17]. Speed, correctness, transparency, trust, efficiency, and security are some of the benefits of SM in the blockchain [18]. Additionally, to further ensure data security, IPFS incorporates an encryption mechanism into the hashes of submitted data. The datasets are encrypted with 256-bit encryption and the returned hash before uploading to IPFS. By accelerating the download of remote sensing data, sharing enormous volumes of data without duplication, and lowering bandwidth costs, IPFS also minimizes storage requirements. An IPFS object, also known as a structure of data and connections, is used to store data in a file. Data files greater than 256 KB are divided up and stored as several IPFS objects, each of which is connected to a distant sensing data file object by a single empty object. Since IPFS enables a hash string route for data transfer and changes to a hash value have an impact, it is commonly used. Decentralizing the datasets and lowering the burden on the data server are made possible by IPFS's higher bandwidth. The entire dataset of cipher text that has been encrypted is uploaded to IPFS storage.

Big Data's era has created enormous obstacles and limitless potential for people all over the world. Big Data analytics has sparked advancements and discoveries in a wide range of fields, including crime [19], causality analysis [20], energy, forecasting [21], and banking [22] to name a few. The banking industry is benefiting from Big Data analytics in terms of security enhancement, risk management, customer relationship management, and marketing, which has greatly improved its operational efficiency and profits, according to the exhaustive evidence in [22]. Cryptocurrencies are intricately tied to Big Data in numerous ways as a recently expanding sector [23]. In this part, two major ideas in the contemporary digital world cryptocurrency and big data—are examined and briefly discussed. Please take notice that [24] contains a recent assessment of blockchain applications in Big Data, however, it only looked at a few up to 2016. In order to provide the most recent overview that holistically outlines the interactions between big data and cryptocurrency, this paper will only provide the most recent scientific advancements made after 2016.

Big Data and cryptocurrencies are convergent in complementary ways. The characteristics of the cryptocurrency network have indicated its importance as a useful Big

Data analytics resource. In a basic blockchain design, for instance, every participant's transaction records are included in the decentralized system, and the data is accurate and well-structured, creating a data-intensive environment that is ideally suited for the use of big data analytics.

Despite being a more mathematically sophisticated and attractive cryptographic method, AES data encryption's main merit is the availability of different key lengths. AES is significantly more secure than DES since you can select a key size of 128 bits, 192 bits, or 256 bits, as opposed to DES's key size of 56 bits. AES is a quick and trustworthy encryption method that protects data from unauthorized access. The following is a mathematical formula for AES [25].

$$f(x) = \sum_{i=0}^{d} c_i \prod_{j}^{d} = 0\ j \neq i \left( \frac{x - p_j}{p_i - p_j} \right) \tag{1}$$

$$w \mapsto \sum_{i=0}^{7} \lambda_i w^{-2^i} \tag{2}$$

*Motivation:*

This article offers a blockchain-aided approach for regulated remote sensing data and document sharing, motivated by the need for a dependable, trustworthy, decentralized document remote sensing data system. The blockchain refers to a decentralized system in which participants do not need to trust one another and may agree on the existence and status of shared data in an untrustworthy situation [11]. Using cryptographic approaches to protect user identification and provide secure transactions by protecting all data transmitted through the chain, this system eliminates record manipulation. Before being added to the chain, each block of the blockchain is individually validated (e.g., consensus by all active members). Smart contracts may be used to manage new user registration requests. We use smart contracts in our suggested solution to create an algorithm that automates the workflow of digital data logic while also allowing for regulated or unregulated data transfer. The smart contract algorithm essentially orchestrates entire communication among different parties (such as developers and approvers) in a decentralized manner.

The contributions of this research can be:

- We provide a blockchain-aided approach for securing remote sensing data through Ethereum smart contracts. The suggested approach mitigates the need for a trusted third-party authenticator.
- We present the main features of our blockchain solution with regard to the entire system design, highlighting key interactions between participants.
- We use a framework to see if the notion is feasible. In order to do this, we built a secure remote sensing data platform system prototype on the Ethereum test network. The associated source codes have been made available on the internet.
- We used test cases to validate functionality and evaluated the proposed framework's capabilities based on the following performance metrics: document uploading and access time, cost of running functions, time to record system events in the blockchain, average block size, and average gas consumption.

The rest of the article is arranged as follows: Section 2 discusses the background and current state of the art. Section 3 discusses the suggested framework's research approach. Section 4 covers the implementation of technologies and methods, as well as the verification and analysis of the proposed system. The evaluation and dangers are described in Section 5. Finally, Section 6 concludes the paper.

## 2. Legacy Evolutions: Existing View and Emerging Technical Challenges

This Section discusses several studies from literature reported on blockchain-aided remote sensing data security and data sharing of digital contents. We have reviewed current solutions in existing literature from abstracts without providing any implementation details.

The Swedish government [16] employed a blockchain-based document management system to register land papers and document the titles of land to digitize the real estate industry. This technique attempts to establish a more reliable and safe mechanism for updating and exchanging documents across stakeholders. This suggested technique may also be used to confirm the identity of a user who has completed a registration on a smart contract-aided system. The complete historical data of a land document may be traced, stored, and confirmed by all included organizations by utilizing the potential of blockchain technology. The model is well-known, although it is still being tested.

Records Keeper [26] suggested a publicly accessible, open source, mineable blockchain ecosystem for record management and document protection based on blockchain technology and sophisticated encryption technique. This company aspires to provide a solid platform for data transfer and authorization that is both safe and reliable. With increased security [6], accessibility to documents across peer groups using the decentralized capacity network tends to be easier. Conventional database systems including Oracle and MySQL do not offer a platform for creating immutable records that cannot be tampered with, as blockchain-based technology does. Records Keeper provides the end user with a rigorous framework for keeping documents over the blockchain which can be validated at any point in time, allowing the user to focus on the specific use case/problem.

Iron Mountain [27], a global corporate organization, uses a specific approach to store, protect and manage a massive amount of digital data. It offers reliable network storage and information management. By integrating blockchain technology, the network entities that are entirely unknown to each other can proceed with trustworthy transactions; keeping the validity of the digital assets into account. Any approved member of this network can easily track the network changes. Conventional approaches such as digital signatures and digital watermarks have been considered to track the progress and integrity of a physical document. However, these approaches arise concerns about the authenticity of digital assets as data can be easily duplicated, tempered, and damaged. Even though, there are multiple methods to authenticate and secure documents; however, these methods rely on a third-party entity which is untrustworthy, unreliable, and expensive.

The author in [28] developed a revolutionary Fairness Consensus Protocol that utilizes conflict graphs, neural networks, and cryptography to enable the network to come to a consensus while preserving the privacy of the user's data. Which transactions are entered onto the distributed ledger will be determined by them. Network fairness could be improved by preventing the same nodes from forming the consensus committee. When all trustworthy nodes have an equal chance of making the final board, there is a fair consensus [29]. A proof-of-accuracy procedure was introduced, and a protocol was suggested by the author in [30]. The suggested protocol contains several aspects, such as coordinator selection, secret generation, part generation and distribution in the network, and participant competition to discover the shares to reconstruct the secret.

Eleks Labs [31] created a unique technique for protecting document transmission utilizing Ethereum to offer safe capacity and transaction for many sorts of legal, financial, and various kinds of sensitive data [32–34]. The organization designed a secure environment in which legal transactions could be performed without the use of a third-party middleman. The created system is a permissionless blockchain, thus anyone may enter the network as a participant and see or perform transactions. The main objective of this system is to ensure a safe storage and transmission method for different types of documents, including personal information, financial papers, legal agreements, and so on. The major goal of creating this system was to assure efficient and secure transactions by removing the requirement for a middleman. In practice, the participants need a certifier who validates the registers, signs, and contents to certify a legal agreement. Because it does not require a notarization authority, blockchain technology is cost-effective. Ethereum-enabled smart contracts are in charge of document maintenance and verification on IPFS. Cryptographic technology can be used to verify the signatures of several parties. The smart contract offers an interface for limiting document accessibility and monitoring alterations. If they have an encryption key, every authorized member in the network can edit existing files and observe the modifications which can be traced down the chain. Authors in [35] stress the necessity to share, update, alter, and replicate scientific and research publication data. The remote sensing research community such as geologists, environmental experts, and data science experts encourage researchers to share their findings in publications. According to Zigmond and Fischer [36], scientific development is dependent not only on research findings but also on authors' willingness to share their findings. The general data sharing rules have been shown to have two main flaws. To begin with, it was the low percentile of journal publication bodies that agreed to establish a rigorous data-sharing policy. Second, policy instructions are frequently unclear and illogical, leading to widespread denial of the policy by journal publishers.

In the area of remote sensing big data, new developments in both computer science and remote sensing research have shown promise [37–43]. The Analysis Ready Data (ARD) that the Committee on Earth Observation Satellites (CEOS) has recommended is a highly processed, useful output. While the aforementioned study has addressed a number of significant big data concerns in remote sensing [44], it is still challenging for researchers to use. The input format of the framework suggested by Sun et al. does not, for instance, match the data formats of ARD and data cube [45].

A phase of literature analysis presents the existing studies to discuss the solutions and limitations. In the literature review phase, we follow the study [46] to perform literature with the most relevant current papers as per the suggestions. After analyzing of literature, we designed the solution and research scope. The rest of the phases are discussed in the methodology section (see Figure 1).

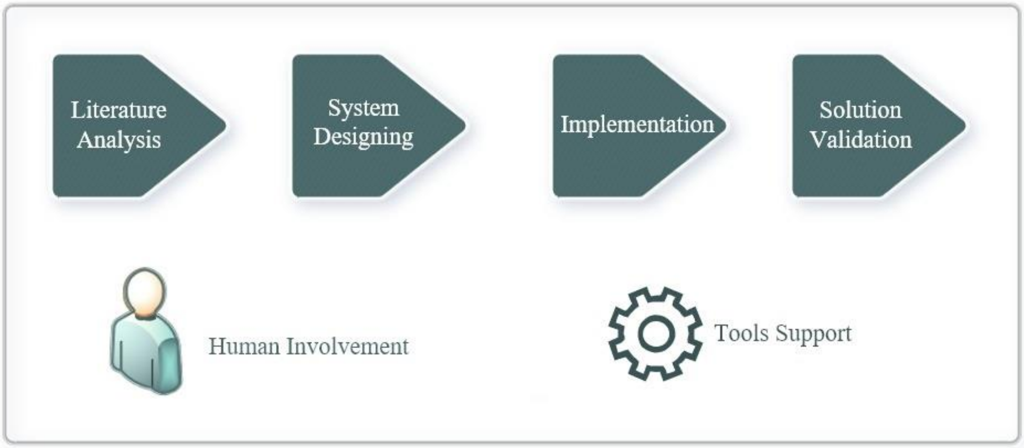

**Figure 1.** Overview of Research Methodology.

The advantages and disadvantages of centralized and decentralized identity management systems are listed in Table 1. We presented four different features to show the difference between central traditional systems and modern blockchain-based systems.

**Table 1.** Comparing Blockchain system vs. traditional system as centralized and decentralized.

| Property | Traditional Systems | Blockchain Systems | Our Scheme |
|---|---|---|---|
| **1-Governance** | Centralized [5,20,36,45] | Decentralized | Decentralized |
| **2-Identity Change** | Easy to change detail on server [5,20,36,45] | Immutable and safe to change history. | Immutable |
| **3-Storage** | Centralized servers [5,20,36,45] | Distributed Nodes. | Blockchain Ledger + IPFS |
| **4-Freedom** | Users' identities are susceptible to theft. | Users recover ownership of their data. | Secure |
| **5-Data Security** | Traditional encryption by third parties [5] | Blockchain Algorithms (SHA256) | AES |

## 3. Research Methodology and Motivation Consequence

In the methodology section, we discuss the flow of study approach including solution design. Figure 1 presents the visualized overview of research techniques with all different phases and approaches for the validation of the solution.

After the first phase of literature, we designed the solution and research scope. The system designing phase is all about solution design and its modeling. We designed different models for our solution. The implementation phase discusses the solution execution in terms of computational usage and storage. Also, discuss the algorithms of the solution. Phase four discussed the validation of the Solution and assesses the functionality and quality of the suggested solution. The model which we utilized is ISO/IEC-9126 [47] in methodology to measure system quality. Using a set of well-established assessment metrics, we focus on evaluating a range of elements of system usability and efficiency.

In Figure 2, we are presenting an overview of the overall system. This system shows all the different processes. Before using this system, the user must be authorized, the users apply for approval to be a part of this system through the registration process. After approval, the user can upload the data and pass the secret key to decrypt the data through the platform. Only authorized users are allowed to get the required data. The second use is an admin who processes the complete data and set the dynamic secret to upload on IPFS.

The ground station network can access the text file containing the binary data that is packaged with the remote sensing satellite data. This bundled data is processed on the data server, where it is encrypted using the AES technique and then transformed into cipher text. The data is saved into a CSV file after encryption and uploaded to an IPFS distributed storage system, which provides the file hash key. By executing the smart contracts, the file hash key and other necessary information are saved in the blockchain ledger. Researchers or other interested parties can access IPFS data by using the created platform as a web server. To access the data, a person must be authorized to view it and have a registered blockchain address. The user accesses the data as cipher texts and transforms it into a readable form using the provided secret key.

Figure 3 presents the process of storing sensing datasets. The data admin uploads the available data of remote sensing to IPFS and gets back the hash key that is stored in the blockchain ledger with other required information. The dataset processes use the symmetric encryption method to encrypt the sensing dataset with a given secret key. After encrypting the dataset into cipher text, the cipher text is written into a file using a file stream library. This encrypted file is uploaded into IPFS and it returns the file hash. The file hash is stored with dataset detail in the blockchain ledger.

Figure 4 presents the process of accessing the documents. The documents owner access the available documents data from IPFS based on the file hash which is stored in the blockchain. The retrieved document is encrypted which needs to be encrypted using the document owner key. It is used to encrypt the document and the same key will be used to decrypt the encrypted document. After downloading the encrypted document cipher text, the cipher text is decrypted and saved to the original readable file.

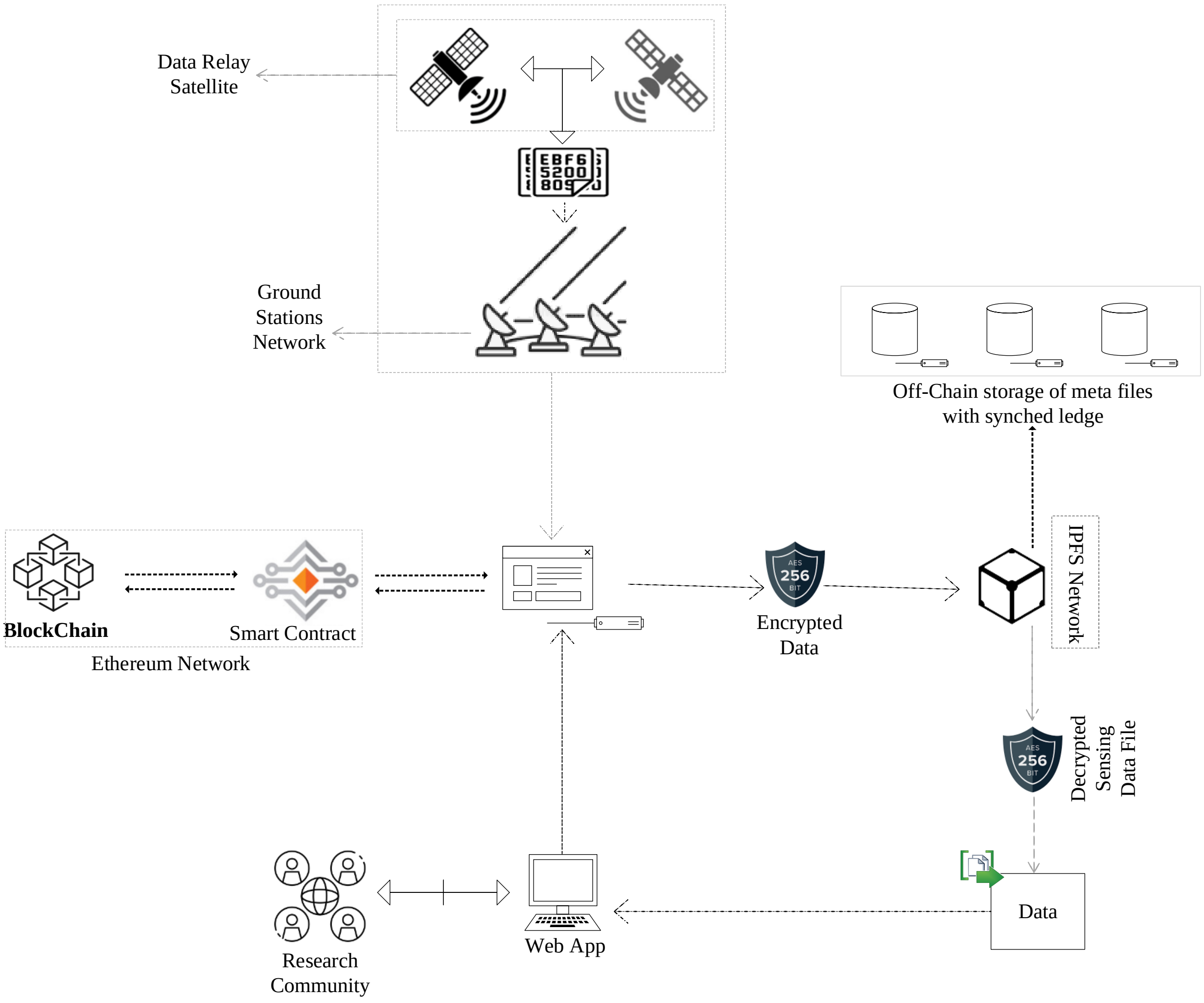

**Figure 2.** Overview of the proposed solution.

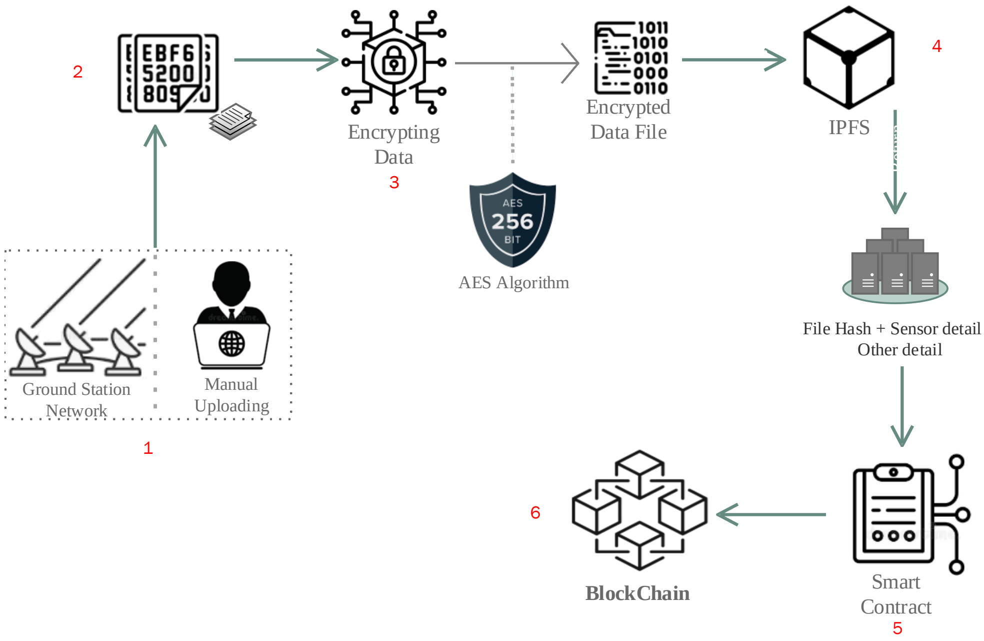

**Figure 3.** Methodology Data saving process.

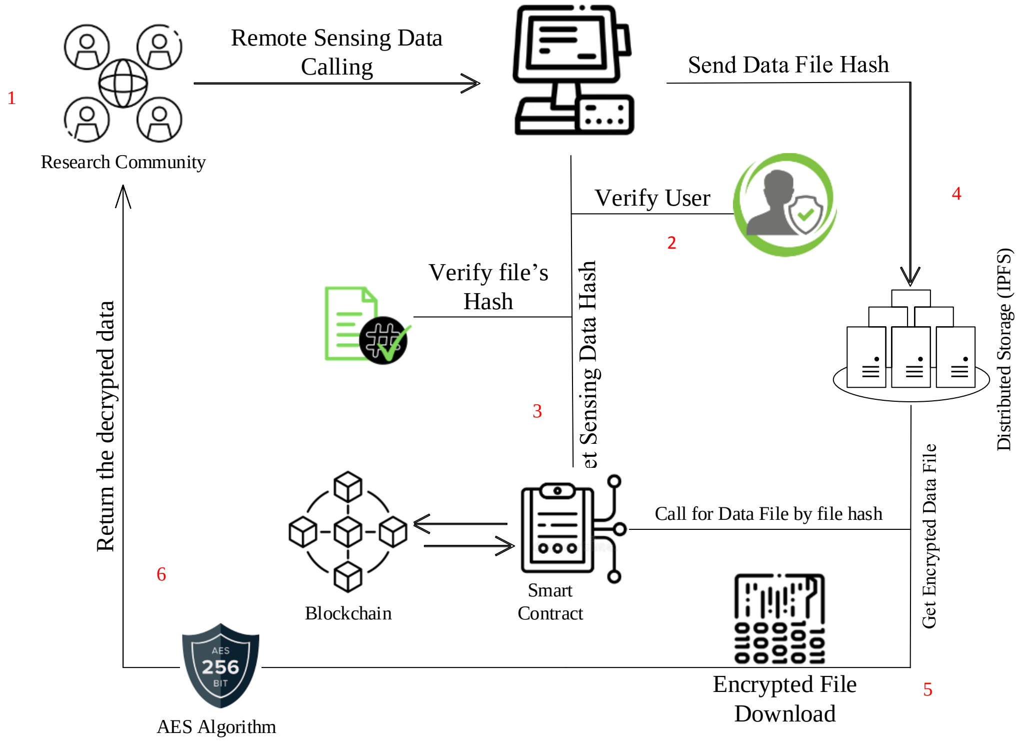

**Figure 4.** Data accessing process.

Figure 5 represents the registration process is the main starting process that is used to give the available roles to a user which could be any role like an admin, approver, or data admin user. The data admin cannot upload the dataset of the remote sensors until getting successful approval from the approver by applying on the registration process.

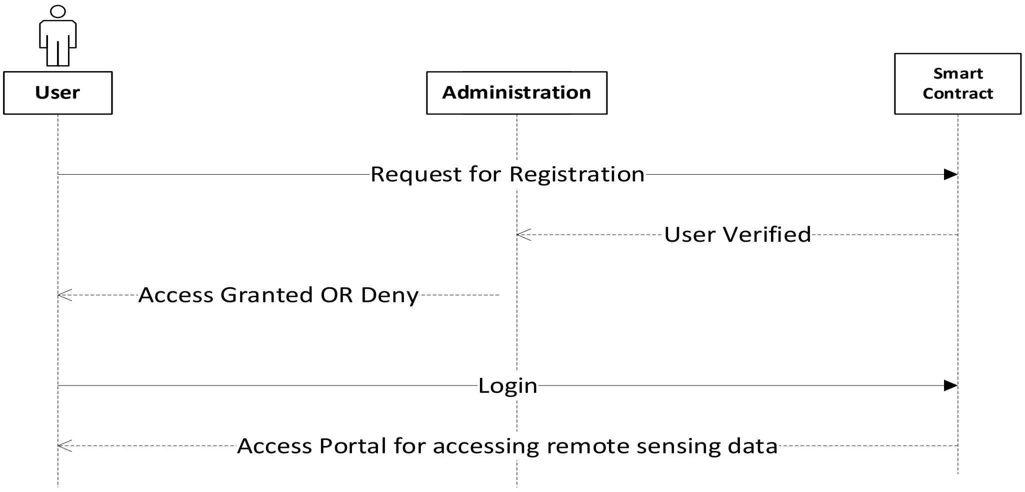

**Figure 5.** The registration process for a user.

The digital data sharing process begins with the creation of metadata for the original file. The remote sensing metadata will comprise data including the file's name, size, description, and type. When the metadata is finished, it is then stored in IPFS with the data file. An example of a file upload to IPFS is as follows:

**Algorithm 1** Pseudo Code

```
IF(function is AddNewRoles)
THEN
{
COUNT + 1;
ADD_RoleId(COUNT);
SET_UserBlockchainAddress(_userAssignToRole);
EMIT_SaveRecordToBlockchainLedger(COUNT, _roleType, _userAssignToRole, _roleStatus, CreatedBlockchainAddress, Date);
}
ELSE IF(function is NewRegistration)
THEN
{
COUNT + 1;
ADD_UserRegistrationId(COUNT);
SET_UserRegistrationBlockchainAddress(CreatedBlockchainAddress);
EMIT_SaveRecordAsNewUserToBlockchainLedger(COUNT,CreatedBlockchainAddress, Date);
}
ELSE IF(function is AddApprovedUser)
THEN
{
COUNT + 1;
ADD_ApprovedUserId(COUNT);
SET_ApprovedUserBlockchainAddress(CreatedBlockchainAddress, COUNT);
EMIT_SaveRecordApprovedUserToBlockchainLedger(COUNT,CreatedBlockchainAddress, Date);
}
ELSE IF(function is SaveRemoteSensingData)
THEN
{
COUNT + 1;
ADD_RemoteSensingDataId(COUNT);
SET_RemoteSensingDataCreatorBlockchainAddress(CreatedBlockchainAddress);
SET_RemoteSensingDataCreatorBlockchainAddressANDid(CreatedBlockchainAddress, COUNT);
EMIT_SaveRemoteSensingDataToBlockchainLedger(COUNT, _remoteSensingId, _locationId, _description, _filehash, _uploadingType,
CreatedBlockchainAddress, Date);

}
```

As per the above Algorithm 1 of "NewRegistration", we send the required parameters to store the data into the blockchain using a smart contract. This function is written in smart contracts using solidity language. We mapped three different mapping sets for searching the data on the portal. The first category is used to get a list of all registered users, while the second mapping is used to get data through the blockchain addresses of the user. We submit the appropriate parameters to save the data into the blockchain using a smart contract utilizing the "AddApprovedUser" function. For finding the data on the site, we created three categories. The first category is used to acquire a list of all approved users, while the purpose of the second mapping is to get data by approval blockchain address with a set of counts. In addition, the third mapping is used to access data for approved users through the user blockchain address. The "Saving" function save the detailed sensing dataset. We divided mapping into three groups. The first mapping is used to get a list of all sensing datasets based, while the second mapping is used to get a dataset for the data admin based on their blockchain address to get the last updated record, and in the third mapping, the admin of the dataset gets the list of the all uploaded datasets based on admin blockchain address and the default set of counts looping.

There are two different symmetric methods in Table 2. The first method is used to encrypt the documents into cipher text and write that cipher text into a file using "FS" library. The second method is used to decrypt the encrypted file which is saved in cipher text.

**Table 2.** Symmetric method source.

```
function encryptSensingFile(setkey, sensingdataFile, encrypteddataFile) {
const inputData = fs.readFileSync(sensingdataFile);
const cipher = crypto.createCipheriv(Algorithm, setkey, Buffer.alloc(16));
const outputcipher = Buffer.concat([cipher.update(inputData), cipher.final()]);
fs.writeFileSync(encrypteddataFile, outputcipher);
}

function decryptSensingDataFile(passkey, sensingdataFile, decrypteddataFile) {
const inputData = fs.readFileSync(sensingdataFile);
const cipher = crypto.createDecipheriv(Algorithm, passkey, Buffer.alloc(16));
const outputcipher = Buffer.concat([cipher.update(inputData), cipher.final()]);
fs.writeFileSync(decrypteddataFile, outputcipher);
}
```

## 4. Implementation of Algorithms and Technologies for Framework

In this section, details on how to implement this algorithm are presented. Ethereum blockchain private network is the suggested system. It is a distributed open-source network that ensures the good utility of Solidity. A programming language that permits smart contracts to be written such as script writing. As a web server, Node.js 15.3.0 was utilized, together with Truffle 5.3.0, Ganache 2.5.4, and IPFS version 33.1.1. For the DApp's networking, we utilized 802.11nWiFi.

### *4.1. Overview of System*

Visual Studio Code: Visual Studio Code (VSC) refers to a code editor by Microsoft that works on a range of operating systems. VSC is a dual-licensed source-code editor for Windows, Linux, and macOS from Microsoft. Debugging tools, highlighted syntax, intelligent code completion, integrated Git control, and code rewriting are all available [48].

Ganache: Ganache is a blockchain-based emulator that can conduct a variety of tests and commands. Ganache is a personal Ethereum blockchain that can be used to run tests, deploy contracts, and construct apps. It inspects the system's statuses and thereby controls the blockchain's functioning. It was once called Test RPC, but it was later renamed ganache [49].

Metamask: It refers to a browser extension that connects to a distributed web. Rather than operating the entire Ethereum node, it runs Ethereum decentralized applications in the browser. To access their Ethereum wallet, users can utilize a browser [50].

IPFS: IPFS is a decentralized open storage system that uses a hash string route to move data. It's used to hold data that's been encrypted and contains additional data. The routes work in a similar fashion to the traditional web's universal resource locator. As a result, all remote sensing data may be retrieved using their hash at any moment.

Algorithm 2 presents the authentication process. All stakeholders need to be authenticated before access of data.

**Algorithm 2** Authentication

```
Input: $\Phi p$                                   Blockchain Address
Output: bool
procedure AUTHENTICATION
if msg.sender is not $\Phi p$ then                Not registered before
    Registration ($\Phi p$)                       Request for approval
end if
        if msg.sender is Valid then               Approved User
    Dashboard()                                   User can upload sensing datasets
end if
end procedure
```

In Algorithm 3, stakeholder authorization is used to check the authorization for remote sensing data access with different roles is demonstrated and described in this section.

Input(s): The parameters are mapped with a file hash key using the algorithm's input.

Processing: The remote sensing data image file is read and converted into a buffer package, which is then posted to IPFS as a remote sensing data file and the hash key is returned. Additional parameters are linked to the hash key of submitted data. User ID, Appointment ID, Description, and Date are entered into a smart contract, and a blockchain is used to store them.

Output: The mapped data is stored in the blockchain as the output.

**Algorithm 3** Authorization

```
Input: $\tau$, $\Phi p$                                  Authorization Type, User Blockchain Address
procedure AUTHORIZATION-THE-USER                          Event based function
      if msg.sender is Valid then                         Verify Role Manager
    if $\Phi p$ not exist then                            If user does not exist in roles
      $\mu$ AddUserToRole($\tau$,
$\Phi p$)                                                 Pass User Role Type & Blockchain Address

        end if
      end if
      Save($\mu$) $d$ Execute Smart Contract to save records
end p
```

In Algorithm 4, the remote sensing data stored in the blockchain ledger feature are demonstrated and described in this section. The technique is used to store data such as aerial imagery, maps, thematic maps, etc. to a blockchain ledger using a smart contract with a mapping of certain additional properties.

Input(s): Using the algorithm's input, the parameters are translated to user id and user appointment id.

Processing: The remote sensing data is subsequently submitted as a data report to the blockchain ledger. For preserving data in the blockchain, additional factors such as user id and user appointment id are connected. User IDs and other parameters are saved in a smart contract on the blockchain.

Output: The mapped data is stored in the blockchain ledger as the output.

**Algorithm 4** Remote Sensing Data Saving

```
Input:      $RS(\imath d)$, $\lambda$, $\Delta p$, $\gamma\wp$, $\tau$       RS ID, Location, Description, Sensing Data File, Uploading Type
Output: R                                                                 Returning Result
procedure SENSINGDATA                                                     Event based function
  if $\tau$ == $SYST\ EM$ | | $\tau$ == $MANUAL$ then
    if $\tau$ == $MANUAL$ then
      if msg.sender is Valid then                                         Check User validation
          $FS \leftarrow$ File($\gamma\wp$)                 Get File stream                    $FS$
          $FB \leftarrow$ $Buffer.form$ ($FS$)                            Convert FS to Buffer FB
                                      $ENCRYPTED \leftarrow AES(KEY, FB)$ Encrypt the File with key
                                      FH ← IPFS.ADD(ENCRYPTED) Get Hash of Sensing Data FH
   SAVE(RS($\imath d$), $\lambda$, $\Delta p$, $\gamma\wp$, $\tau$ )
        end if
      end if
      if $\tau$ == SYSTEM then
          FS ←File ($\gamma\wp$)                                          Get File stream FS
          FB ←$Buffer.form$ (FS)                                          Convert to Buffer
        ENCRYPTED ← AES(KEY,$FB$)                                         Encrypt the File with key
        $FH \leftarrow IPFS.ADD$(ENCRYPTED)                               Get Hash of Sensing Data
    SAVE(RS($\imath d$), $\lambda$, $\Delta p$, $\gamma\wp$, $\tau$) $d$ Store Data to Blockchain with file hash
    end if
  end if
end procedure
```

Algorithm 5 validates the data accessing capabilities, which are then given in this section. The algorithm is used to obtain data from the blockchain and make it publicly visible. The data from the blockchain may be accessed by the user based on the parameters set. There are several sorts of data access, for example, a user can access data based on their user id and appointment id mapping. Remote sensing specialists can access the stored data directly by user appointment id.

Input(s): The settings for accessing the data are mapped using the algorithm's input.

Processing: The data from blockchain might be accessible in a variety of ways, such as by a user id mapped to an appointment id, or by a geologist accessing geographic image data by a user appointment id.

Output: The result is publicly accessible data that has been mapped.

**Algorithm 5** Interface Layer

```
Input: $\gamma\wp$, $\Delta p$                                  File Hash ID, User Blockchain Address
Output: R                                                        Display Data
    procedure ACCESSING-REMOTE-SENSING-DATA                       Event based function
    if msg.sender($\Delta p$) is Valid then                      If User is Valid
        $FH \leftarrow$ GetFileHash($\gamma\wp$)                 Get File Hash
        $ENCRYPTED \leftarrow$ IPFS($FH$)                         Download Encrypted File based on Hash
        $DECRYPTED \leftarrow$ AES($KEY, ENCRYPTED$)              Decrypted the Encrypted
        $R \leftarrow$ DOWNLOAD($DECRYPTED$)                      Return Sensing Data File
    end if
UpdateDashboard(R)                                                Show Remote Sensing Datasets List
end procedure
```

### 4.2. Tools and Technologies for Algorithmic Implementation

The complementary function of relevant tools and technology for the suggested solution is summarized in this section. The purpose of this discussion is to empower the reader with a better knowledge of technology. If the portal user is a radiologist, the data may be a remote sensing data file that is submitted in encrypted form to the IPFS platform and returned as a hash key. A server-side application is built using the NodeJS platform, which contains a number of tools. To launch the NodeJS application, we used Visual Studio Code (VSC). To construct a local Blockchain environment, we used the Ganache Truffle Suite package to quickly establish a personal Ethereum blockchain that you can use to run tests, issue commands, and observe the state while managing how the chain runs.

## 5. Evaluations and Validity Threats

In this part, the results of the recommended solution are reported. The assessment environment is first, followed by a fuel usage-based evaluation of smart contract functionality. Following that, we use criteria to assess and quantify data uploading and storage to the blockchain, response to inserted query such as performance, and algorithmic execution such as efficiency. The assessment criteria are based on the ISO/IEC-9126 model, which is intended to assess software-intensive systems' quality. Threats to the research's validity, as well as any limitations that must be addressed, are also discussed.

### 5.1. Evaluation Environment

The hardware and software resources are part of the evaluation environment as a collection for the solution of outcomes. The technologies are used in this evaluation environment mentioned below.

Hardware

- Operating System: Windows 10
- Runtime Memory: 16 GB
- CPU: 2.5
- Core: i7

Software

- Tool: Visual Studio Code
- Platform: NodeJS
- Language: ReactJS and some frontend languages
- Libraries: React, Web3, Ipfs.http, JavaScript library which is used to analyze the CPU consumption.
- Extension: The Ganache suit for the cost of the gas transaction to be performed system connects local Ether accounts with Metamask plugin.

### *5.2. Data Uploading and Fuel Consumption*

The fuel is utilized to carry out Ethereum's smart contracts. The Gwei is referred to as the smallest unit of the Ether price for tracking fuel usage in the Ether cryptocurrency. The cost of contract migration execution was specified in our suggested solution (see Table 3). The price is given in Ether, and the gas spent is noted. Ether is equal to the amount of gas utilized multiplied by the price of gas. The gas reflects the continual computing cost in this system. The gas price has been changed by the network [51] to account for changes in the value of Ether.

**Table 3.** Cost analysis for executing smart contracts (gas cost = 2 Gwei).

| Execution Type | Gas Used | Cost in Ether |
|---|---|---|
| Contract Creation | 2,869,227 | 0.05738454 |
| Contract Migration Call | 27,363 | 0.0054726 |
| Initial Contract | 225,237 | 0.0450474 |
| Initial Migration Call | 42,363 | 0.0084726 |
| Final cost | - | 0.06188928 |

We established a gas restriction by default in the implemented prototype of our system. The Contract is created once at a cost of 0.05738454 Ether, with a total gas use of 2,869,227. The migration necessitates Contract formation at a low cost of 0.0054726 (Ether) and gas consumption of just 27,363. If the amount of the input data is kept to a minimum, the overall expenses can be further reduced.

The time it took for users to upload and store data to IPFS and blockchain ledger was the final test item. The overall time spent uploading remote sensing data, recalling available data, and reviewing data is referred to as data uploading and accessing time. Figure 6 shows the outcomes of a series of experiments with average data size. While uploading data of 450 bytes, the average fuel consumption is around 555,062 Gas, and when storing data of 1000 bytes, the average fuel consumption is around 1,409,568 Gas. This demonstrates that as the size of the data grows, so does the amount of gasoline consumed. However, even though the data amount increased, there was no significant difference in fuel use when remote sensing data was uploaded to IPFS using the suggested methodology.

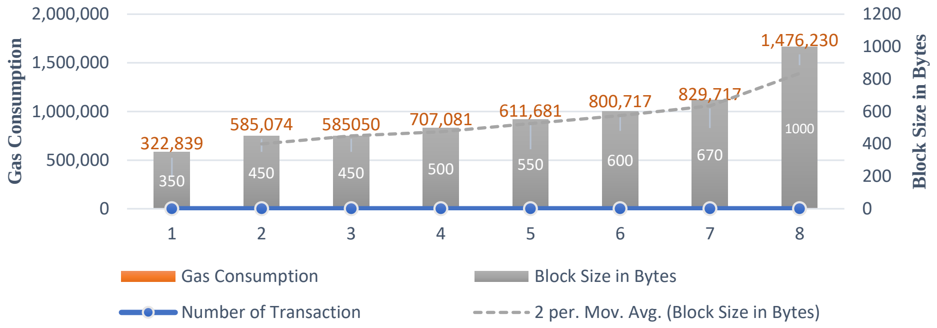

**Figure 6.** Depending on block size and transaction volume, gas is used.

The sequence diagram for all of the network's entities and their interactions is shown in Figure 7. The system's execution flow is visualized in Figure 7 to show how it works. There are six entities (Stakeholders, Remote Sensing, Web, AES, IPFS, Smart Contract). Stakeholders are used to accessing the data from the blockchain ledger using the web portal. Web server entity is used to upload the remote sensing data to IPFS and get back file hash which is mapped with other required info parameters and stored in blockchain.

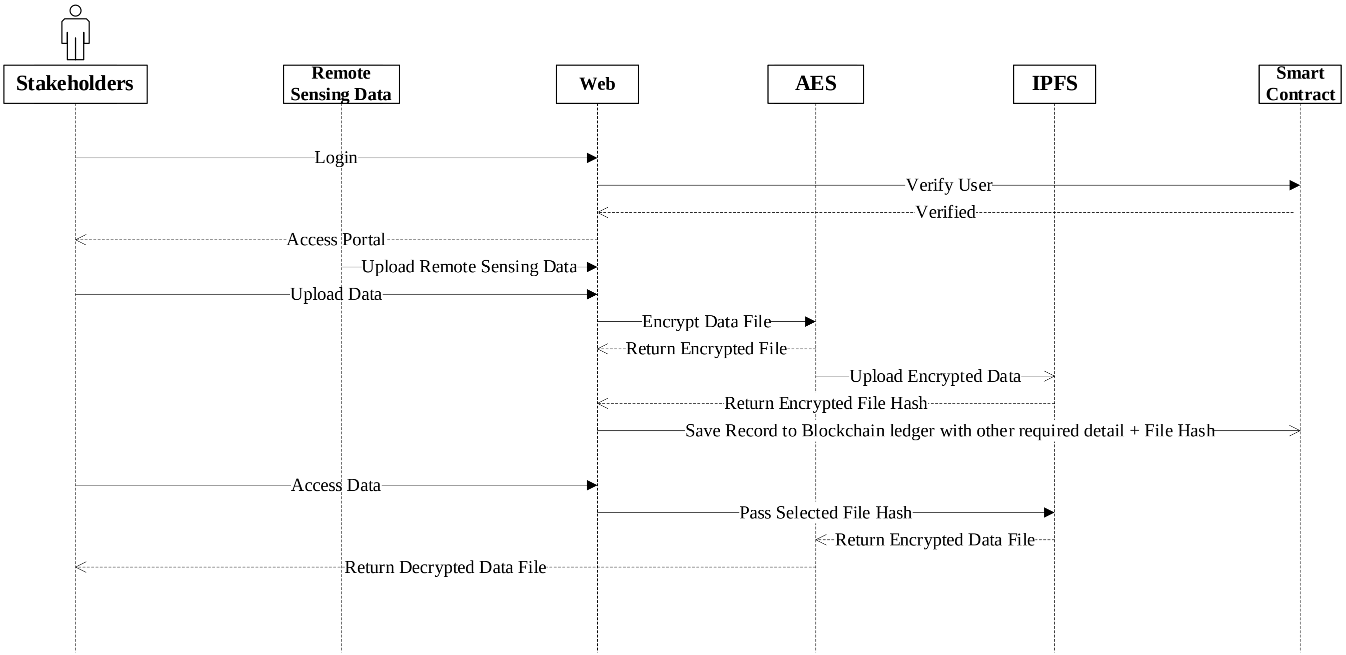

**Figure 7.** System sequence process for remote sensing datasets.

1. Stakeholders: it can be any end user like the research community who needs to get scientific data for their use, the industrial entity for their business perspective use, or for a normal user.
2. Remote Sensing Data: this module is used to gather the data and package all binary data as per requirements. After packaging all data is sent to the database server where all these are processed.
3. Web: this is the presentation module where all the stakeholders are connected and used to get the data. This module is interlinked with IPFS and internal algorithms which we used in the system.
4. AES: it is an algorithm to secure the data by converting it into cipher text as an encryption method. We used 256 bits which allows more length to create a dynamic secret key for securing the data in cipher form.
5. IPFS: this module is for decentralizing all the data packaged which comes from the remote sensing, after encryption the data package is saved into IPFS and returns the file hash key for that specific dataset.
6. Smart Contracts: are used to store and record each transaction in a blockchain ledger. In our scenario, smart contracts are executed after uploading the dataset to IPFS and getting the file hash key which is recorded in the blockchain ledger with other required parameters using smart contracts.

### *5.3. Evaluations Response Time*

Remote sensing data has been saved using IPFS, a distributed storage system built on the blockchain, and transactions with the basic essentials of information are recorded in the blockchain ledger. We assessed the solution's query response time to determine how quickly it could save and retrieve data from a blockchain ledger. We ran two distinct tests to determine how quickly data saved to IPFS and blockchain ledgers responded to queries.

The query results are shown in milliseconds for query response time in Figure 8. The axis on the horizontal shows the execution time, and the vertical shows the response time. The "Complete function" shows the execution of the entire process, from storing remote sensing data to IPFS to saving the record information to the blockchain with the remote sensing data file hash. The "Smart Contract Function" displays the delay caused by Metamask's Smart Contract execution call.

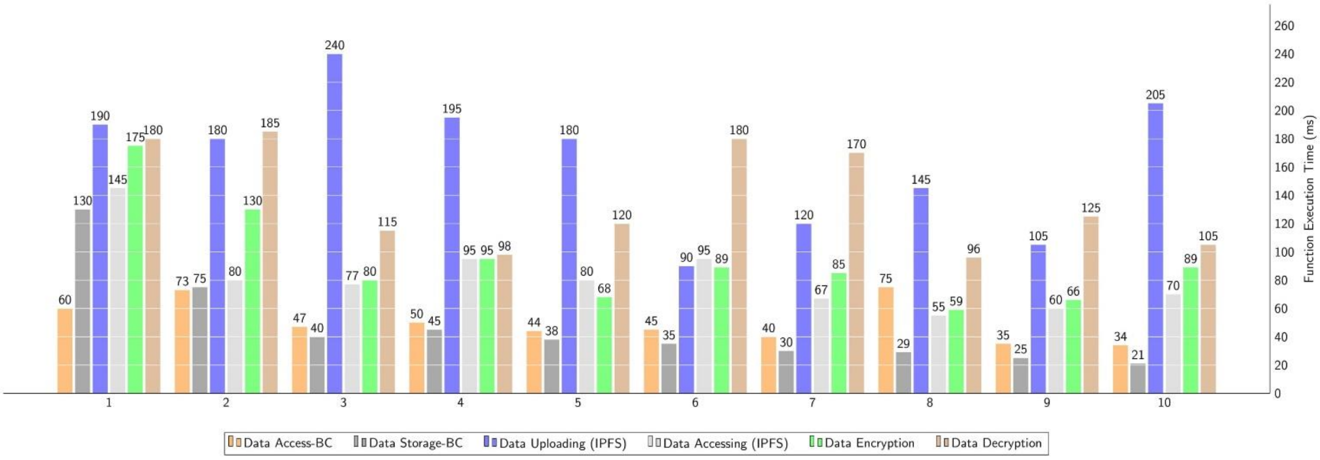

**Figure 8.** The computational time for function execution to store and access the data in IPFS and Blockchain.

The data is coming in two different types. The first type of data is remote sensing which comes from the IPFS through file hash. But number 6 the execution time is high because here the file size has increased, it is a big size remote sensing data that getting more time to show or download from the IPFS. The second part is to get the textual reports of blood or lipid from the blockchain ledger.

### 5.4. Threats to Validity

There are few issues of validity have been discussed. In other words, the validity is also called a limitation that affects the validation and implementation of the solution. Validity must be eliminated as part of future attempts to enhance the solution and its repercussions.

Threats to Internal Validity: it discusses the impact of restrictions or limitations for the proposed system and its implementation. For example, if remote sensing data is utilized to execute the trails in order to obtain the output, the outcome may change in terms of performance.

External Validity: it talks about case studies and related systems for validation of the solution. We employed a case study strategy, as detailed in the research method and assessment section, to demonstrate and evaluate the answer. In the future, more case studies will be required to lessen the impacts of external validity.

## 6. Conclusions

In this paper, blockchain technology is briefly introduced, and the application prospects of its technological advantages of remote sensing data security. This study proposes a decentralized platform for securing the big data of remote sensing. Our proposed approach leverages the advantages and potentials of IPFS file system, smart contracts, and Blockchain. The proposed mechanism is resilient, secure, and entirely decentralized and it eliminates the dependency on the trusted third party. The implementation and testing of the functionalities of the proposed solution have been carried out considering Remix IDE. The proposed framework of this study with the system design, smart contract code, and algorithm is general and can be modified to relevant systems and access to shared digital libraries and data including photos, audio, and video.

We used an experimental implementation to study and assess the suggested scheme's efficiency, rationality, and practicality. While sharing access to user, the suggested system provides encrypts the data to an immutable IPFS, resulting in increased efficiency, data provenance, and effective audit and security from malicious. Because the data storage and exchange mechanism are decentralized, there is no need for third-party middlemen or administrative organizations.

Innovative points are achieved:

- Securing remote sensing data and mitigate the need of third-party authentication
- We designed the different processes to develop the proposed solution
- We developed a platform to perform the prototypes on the Ethereum test network using smart contracts.

*Future work:* In the future, we'll concentrate mostly on the variety of data evaluation with more case studies, which can further increase the evaluation's rigor. Additionally, the approach will be enhanced with signature verification and put into practice using remote sensing on oceanic data.

**Author Contributions:** Conceptualization, A.R. and S.A.H.M.; methodology, A.R., S.A.H.M. and S.A.K.G.; resources, M.H.A. and S.M.M.; validation, S.M.M., H.K.A. and F.K.K.; writing—original draft preparation, A.R., S.A.H.M. and M.H.A.; writing—review and editing, M.H.A., S.M.M., H.K.A. and F.K.K.; Supervision, M.H.A.; funding acquisition, F.K.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research project was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R300).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Alsharif, M.H.; Kelechi, A.H.; Albreem, M.A.; Chaudhry, S.A.; Zia, M.S.; Kim, S. Sixth Generation (6G) Wireless Networks: Vision, Research Activities, Challenges and Potential Solutions. *Symmetry* **2020**, *12*, 676. [CrossRef]
2. Alsharif, M.H.; Hossain, M.S.; Jahid, A.; Khan, M.A.; Choi, B.J.; Mostafa, S.M. Milestones of Wireless Communication Networks and Technology Prospect of Next Generation (6G). *Comput. Mater. Contin.* **2022**, *71*, 4803–4818. [CrossRef]
3. Rashvand, H.F.; Salah, K.; Calero, J.M.A.; Harn, L. Distributed security for multi-agent systems-review and applications. *IET Inf. Secur.* **2010**, *4*, 188–201. [CrossRef]
4. Lv, Z.; Chen, D.; Lv, H. Smart City Construction and Management by Digital Twins and BIM Big Data in COVID-19 Scenario. *ACM Trans. Multimed. Comput. Commun. Appl.* **2022**. [CrossRef]
5. Cao, B.; Zhang, J.; Liu, X.; Sun, Z.; Cao, W.; Nowak, R.M.; Lv, Z. Edge-Cloud Resource Scheduling in Space-Air-Ground Integrated Networks for Internet of Vehicles. *IEEE Internet Things J.* **2021**, *9*, 5765–5772. [CrossRef]
6. Wu, X.; Zheng, W.; Xia, X.; Lo, D. Data Quality Matters: A Case Study on Data Label Correctness for Security Bug Report Prediction. *IEEE Trans. Softw. Eng.* **2021**, *48*, 2541–2556. [CrossRef]
7. Yang, L.; Xiong, Z.; Liu, G.; Hu, Y.; Zhang, X.; Qiu, M. An Analytical Model of Page Dissemination for Efficient Big Data Transmission of C-ITS. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16524–16533. [CrossRef]
8. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://bitcoin.org/bitcoin.pdf (accessed on 15 June 2022).
9. Khan, M.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
10. Salah, K.; Rehman, M.H.U.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and Open Research Challenges. *IEEE Access* **2019**, *7*, 10127–10149. [CrossRef]
11. Zhao, J.L.; Fan, S.; Yan, J. Overview of Business Innovations and Research Opportunities in Blockchain and Introduction to the Special Issue. *Financ. Innov.* **2016**, *2*, 1. [CrossRef]

12. Scherer, M. Performance and Scalability of Blockchain Networks and Smart Contracts. 2017. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1111497&dswid=-3678 (accessed on 20 June 2022).
13. Bocek, T.; Rodrigues, B.B.; Strasser, T.; Stiller, B. Blockchains Everywhere-A Use-case of Blockchains in the Pharma Supply-Chain. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 772–777.
14. Daniel, F.; Guida, L. A service-oriented perspective on blockchain smart contracts. *IEEE Internet Comput.* **2019**, *23*, 46–53. [CrossRef]
15. Leng, J.; Ruan, G.; Jiang, P.; Xu, K.; Liu, Q.; Zhou, X.; Liu, C. Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey. *Renew. Sustain. Energy Rev.* **2020**, *132*, 110112. [CrossRef]
16. Rizzo, P. Sweden Tests Blockchain Smart Contracts for Land Registry. Available online: https://www.coindesk.com/sweden-blockchain-smart-contracts-land-registry/ (accessed on 20 May 2018).
17. Benet, J. Ipfs-content addressed, versioned, p2p file system. *arXiv* **2014**, arXiv:1407.3561.
18. Puri, V.; Priyadarshini, I.; Kumar, R.; Van Le, C. Smart contract based policies for the internet of things. *Clust. Comput.* **2021**, *24*, 1675–1694. [CrossRef]
19. Hassani, H.; Huang, X.; Silva, E.S.; Ghodsi, M. A review of data mining applications in crime. *Stat. Anal. Data Min. ASA Data Sci. J.* **2016**, *9*, 139–154. [CrossRef]
20. Hassani, H.; Huang, X.; Ghodsi, M. Big Data and Causality. *Ann. Data Sci.* **2017**, *5*, 1–24. [CrossRef]
21. Hassani, H.; Silva, E.S. Big Data: A big opportunity for the petroleum and petrochemical industry. *OPEC Energy Rev.* **2018**, *42*, 74–89. [CrossRef]
22. Hassani, H.; Huang, X.; Silva, E.S. Digitalisation and Big Data Mining in Banking. *Big Data Cognit. Comput.* **2018**, *2*, 18. [CrossRef]
23. Chuen, D.L.K. (Ed.) *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*; Academic Press: Cambridge, MA, USA, 2015.
24. Karafiloski, E.; Mishev, A. Blockchain solutions for big data challenges: A literature review. In Proceedings of the IEEE EUROCON 2017—17th International Conference on Smart Technologies, Ohrid, Macedonia, 6–8 July 2017.
25. Razzaq, A. Blockchain-based secure data transmission for internet of underwater things. *Clust. Comput.* **2022**. [CrossRef]
26. RecordsKeeper. Available online: https://www.recordskeeper.co/ (accessed on 1 July 2021).
27. Iron Mountain. Available online: http://www.ironmountain.com/resources/general-articles/w/what-is-blockchain-and-why-should-records-management-professionals-care (accessed on 5 July 2021).
28. Caldarola, F.; d'Atri, G.; Zanardo, E. Neural Fairness Blockchain Protocol Using an Elliptic Curves Lottery. *Mathematics* **2022**, *10*, 3040. [CrossRef]
29. Yan, L.; Yin-He, S.; Qian, Y.; Zhi-Yu, S.; Chun-Zi, W.; Zi-Yun, L. Method of Reaching Consensus on Probability of Food Safety Based on the Integration of Finite Credible Data on Block Chain. *IEEE Access* **2021**, *9*, 123764–123776. [CrossRef]
30. Aponte-Novoa, F.A.; Villanueva-Polanco, R. On Proof-of-Accuracy Consensus Protocols. *Mathematics* **2022**, *10*, 2504. [CrossRef]
31. Eleks. Secure Document Transfer Built on Top of Blockchain Technologies. Available online: https://labs.eleks.com/ (accessed on 1 July 2021).
32. Zheng, H.; Jin, S. A Multi–Source Fluid Queue Based Stochastic Model of the Probabilistic Offloading Strategy in a MEC System with Multiple Mobile Devices and a Single MEC Server. *Int. J. Appl. Math. Comput. Sci.* **2022**, *32*, 125–138. [CrossRef]
33. Wang, Y.; Han, X.; Jin, S. MAP based modeling method and performance study of a task offloading scheme with time-correlated traffic and VM repair in MEC systems. *Wirel. Netw.* **2022**. [CrossRef]
34. Shen, Y.; Ding, N.; Zheng, H.T.; Li, Y.; Yang, M. Modeling Relation Paths for Knowledge Graph Completion. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3607–3617. [CrossRef]
35. Vasilevsky, N.A.; Minnier, J.; Haendel, M.A.; Champieux, R.E. Reproducible and reusable research: Are journal data sharing policies meeting the mark? *PeerJ* **2017**, *5*, e3208. [CrossRef]
36. Fischer, B.A.; Zigmond, M.J. The essential nature of sharing in science. *Sci. Eng. Ethics* **2010**, *16*, 783–799. [CrossRef]
37. Li, Y.; Du, L.; Wei, D. Multiscale CNN Based on Component Analysis for SAR ATR. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5211212. [CrossRef]
38. Liao, L.; Du, L.; Guo, Y. Semi-Supervised SAR Target Detection Based on an Improved Faster R-CNN. *Remote Sens.* **2021**, *14*, 143. [CrossRef]
39. Tian, H.; Qin, Y.; Niu, Z.; Wang, L.; Ge, S. Summer Maize Mapping by Compositing Time Series Sentinel-1A Imagery Based on Crop Growth Cycles. *J. Indian Soc. Remote Sens.* **2021**, *49*, 2863–2874. [CrossRef]
40. Tian, H.; Wang, Y.; Chen, T.; Zhang, L.; Qin, Y. Early-Season Mapping of Winter Crops Using Sentinel-2 Optical Imagery. *Remote Sens.* **2021**, *13*, 3822. [CrossRef]
41. Tian, H.; Huang, N.; Niu, Z.; Qin, Y.; Pei, J.; Wang, J. Mapping Winter Crops in China with Multi-Source Satellite Imagery and Phenology-Based Algorithm. *Remote Sens.* **2019**, *11*, 820. [CrossRef]
42. Zhao, M.; Zhou, Y.; Li, X.; Cheng, W.; Zhou, C.; Ma, T.; Huang, K. Mapping urban dynamics (1992–2018) in Southeast Asia using consistent nighttime light data from DMSP and VIIRS. *Remote Sens. Environ.* **2020**, *248*, 111980. [CrossRef]
43. Zhao, M.; Zhou, Y.; Li, X.; Zhou, C.; Cheng, W.; Li, M.; Huang, K. Building a Series of Consistent Night-Time Light Data (1992–2018) in Southeast Asia by Integrating DMSP-OLS and NPP-VIIRS. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 1843–1856. [CrossRef]

44. Hu, Y.; Qing, J.X.; Liu, Z.H.; Conrad, Z.J.; Cao, J.N.; Zhang, X.P. Hovering efficiency optimization of the ducted propeller with weight penalty taken into account. *Aerosp. Sci. Technol.* **2021**, *117*, 106937. [CrossRef]
45. Sun, J.; Zhang, Y.; Wu, Z.; Zhu, Y.; Yin, X.; Ding, Z.; Wei, Z.; Plaza, J.; Plaza, A. An efficient and scalable framework for processing remotely sensed big data in cloud computing environments. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4294–4308. [CrossRef]
46. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering a systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
47. Estdale, J.; Georgiadou, E. Applying the iso/iec 25010 quality models to software product. In *European Conference on Software Process Improvement*; Springer: Berlin/Heidelberg, Germany; pp. 492–503.
48. Truffle Suite. Available online: https://www.trufflesuite.com/guides/configuring-visual-studio-code.html (accessed on 15 March 2021).
49. Truffle Suite. Available online: https://truffleframework.com/docs/ganache/overview (accessed on 15 March 2021).
50. MetaMask. Available online: https://metamask.io/ (accessed on 15 March 2021).
51. Wood, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Available online: https://gavwood.com/paper.pdf (accessed on 20 March 2021).