Covid-19 Data Sharing and Organization through Blockchain and Decentralized Models

Mario Casillo[1], Arcangelo Castiglione[2], Francesco Colace[1], Massimo De Santo[1], Francesco Marongiu[1] and Domenico Santaniello[1][0000-0002-5783-1847]

[1] DIIn, University of Salerno, Fisciano (SA), Italy
{mcasillo; fcolace; desanto; fmarongiu; dsantaniello}@unisa.it
[2] DI, University of Salerno, Fisciano (SA), Italy
arcastiglione@unisa.it

Abstract. The pandemic due to Covid-19 has highlighted and sometimes amplified some critical issues affecting the global health system. From the beginning, it was clear that having an overview of the pandemic situation was one of the main objectives to be achieved to fight the Virus. In this sense, technologies based on decentralized systems, such as Blockchain, can be effective in collecting data since they can store information in a secure and scalable way while respecting people's privacy.

This work aims to propose a framework based entirely on decentralized systems that can solve management and collaborative analysis of sensitive data, such as those related to the Coronavirus pandemic. In particular, the model is proposed as a tool to improve the efficiency of all those organizations that have found themselves on the front line to deal with the spread of the Virus in the world, such as hospitals and research institutes. In addition, it aims to improve communications between organizations at the global level to facilitate a fair and profitable exchange of knowledge to fight the pandemic.

Keywords: Big Data, Blockchain, Digital Storage, Network Security, Smart Contract, Trusted Data Sharing.

1 Introduction

Data nowadays represent one of the significant sources of knowledge, as their correlations can highlight some information that would otherwise be hidden [1]. This aspect is especially evident in managing a global pandemic: the number of infections, geographic location, patient information, etc., are crucial data in the fight against the pandemic [2] [3].

Knowing precisely the movement of the Virus allows preventing its spread by acting in the most opportune areas and moments [4]. However, the processing of all this information poses many problems, both from the point of view of collection and access and, above all, to protect the privacy of individuals [5].

Another critical aspect to consider is undoubtedly exchanging this information globally, securely, and in real-time, because a situation like the Covid-19 pandemic

can only be solved through appropriate collaborations both locally and internationally [6].

Using a decentralized approach, it is possible to develop secure and scalable software that can accommodate the vast amount of information globally while maintaining a high degree of reliability [7]. Data are therefore distributed within a global network that can maintain unaltered properties and guarantee the security of the information. This result can be achieved by using systems based on Peer-2-Peer networks, in which the exchange of information and documents takes place in a decentralized and autonomous way among the different nodes of the network [8].

Blockchain and [8] decentralized file system technologies (i.e. InterPlanetary FileSystem - IPFS), are well suited to develop architectures that meet the requirements in analysis [9] [10]. The aim of the paper is, therefore, to develop a framework able to guarantee:

- *Security against any accidental and non-accidental disaster*: Blockchain along with decentralized storage (such as IPFS) can ensure the integrity and immutability of data permanently and offer quick access to information worldwide.
- *Privacy*: Using modern cryptographic techniques, it is possible to guarantee a system of data ownership, with the possibility of obscuring sensitive information from the Public through selective access. This aspect represents a significant advantage for people and public and private organizations collecting and working on those data.
- *Real-time sharing* of information. This goal is achieved by encouraging collaboration across multiple organizations and maintaining control over data ownership and viewing.

From the earliest stages of the pandemic, Hospitals and Research Institutes found themselves working with data that was peculiar in quantity and nature. Traditional systems had to be reorganized to try to meet the main requirements of privacy and data size. However, some necessary properties are challenging to achieve with traditional tools. An example is the collection and certification of infections, where sensitive information must somehow be shared worldwide while preserving user privacy [11].

2 Related Works

Several approaches have been proposed in the literature for real-time health monitoring of patients.

Azaria et al. [12] propose MedRec, a decentralized record management system to handle electronic medical records using blockchain technology. MedRec provides capabilities for managing authentication, confidentiality, accountability, and data sharing. The system allows users to control their medical information through a blockchain network consisting of several medical stakeholders securing the network utilizing a Proof-of-Work consensus scheme.

Christodoulou et al. [13] focused on people, giving them exclusive ownership of the data. Using cryptographic techniques and Smart Contracts, they proposed a decentralized system for direct information exchange between doctors and patients by placing the User at the information exchange center.

Such approaches, however, do not apply to a real-world case like a global pandemic. Indeed, to prevent contagions and disease spread, certified and competent Institutions must have complete control over the data to provide the best possible instructions, this is not possible in [12] and [13].

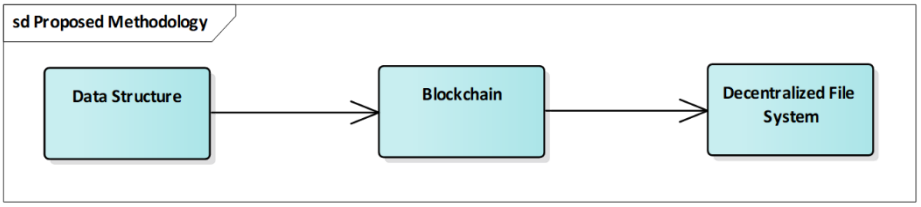
Hasan et al. [14] proposed an interesting approach in which IPFS and Blockchain events are used. In this model, data is selectively encrypted and shared using an asymmetric key encryption mechanism. However, the use of the Blockchain is marginal as only events are utilized while storing any data is given over to IPFS. The letter does not include itself to spread data across the network to decentralize and secure; this can only be done using FileCoin[1], thus additional costs. Finally, their proposed sharing system requires the use of a module dedicated to exchanging keys and references to files; the presence of this module partially reduces the effectiveness of Blockchain advantages.

Instead, the proposed solution aims to combine Blockchain and decentralized storage, trying to benefit from the best features of both.

- The Blockchain is used to store metadata of information in a secure, immutable, and sequential way.
- Decentralized storage is used to store appropriately encrypted information in a scalable way without impacting Blockchain performance.
- In addition, the Blockchain takes care of the access and sharing of metadata among the different users of the system without the intervention of other external modules.

3 Proposed Methodology

The framework is proposed as a general solution for the problem of storing a large amount of data in a secure, traceable, and immutable way, which at the same time respects the confidentiality of the information and facilitates collaboration in the study and analysis of data. For this reason, the system is based on three main concepts: Blockchain; Decentralized Storage; Symmetric and Asymmetric Key Encryption [15].



[1] https://docs.filecoin.io/about-filecoin/ipfs-and-filecoin/#data-storage-incentives

Fig. 1. Modules of the proposed methodology.

The first module involves organizing data into a common and well-defined structure; the second uses blockchain technology for storing file meta-data, the third and final module stores data on the decentralized file system.

For experimental purposes, the model has been implemented using the Ethereum Blockchain and EVM (**E**thereum **V**irtual **M**achine) [16] for the part concerning the Smart Contract. The decentralized file system module was instead implemented using IPFS.

The different modules of which the architecture is composed will be specified in detail below.

For convention, the generic term *User* will be referred to the main actor of the system. In this particular use case, a User will be a single organization which have to store and organize massive amount of data collected by its infrastructure.

3.1 Data Structure

Before any information can be saved, a data collection standard is first defined. This step is crucial, as bringing the data back into one standard structure facilitates all subsequent operations [17]. Especially when it comes to health data, it is not always possible to have a fixed information structure.

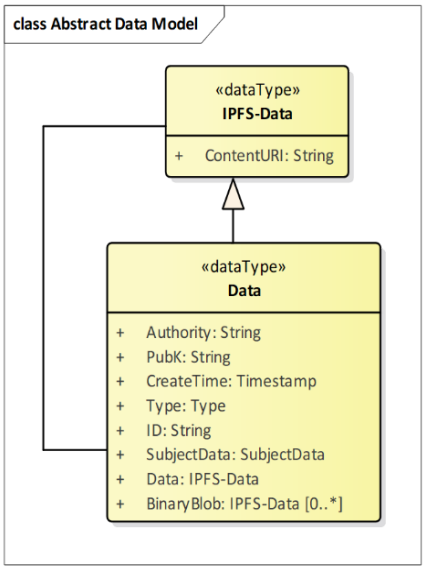


Fig. 2. Abstract Data Structure definition

Due to the very nature of the data, there can be indefinite variations of a common problem or disease. Therefore, a tree structure was created, in which each node represents the smallest information unit, which is desired to be saved immutably.

Each piece of data is uniquely identified by a URI (**ContentURI**) that each User arbitrarily assigns to their personal storage space. Using this structure, it is possible to describe information without a predefined data schema dynamically. In addition, the *ContentURI* field allows information to be structured semantically as well.

3.2 Blockchain

The Blockchain contains all the information regarding the file's metadata. In particular: the date of insertion; the owner (the User who uploads the file); whether the data can change over time; and finally, the reference to the real location of the information on the decentralized file system. Only the meta-data of the information was chosen to be saved on Blockchain because, due to their structure, these systems are not scalable for storing large files [18]; moreover, operations on Blockchain are expensive both in economic and computational terms.

This module was implemented by using the concept of Smart Contract [19], which allowed the secure storage of information, but the Smart Contract itself acts as a point of information interchange between the various users. In this way, no other module is needed for communication, but the parties involved interact independently.

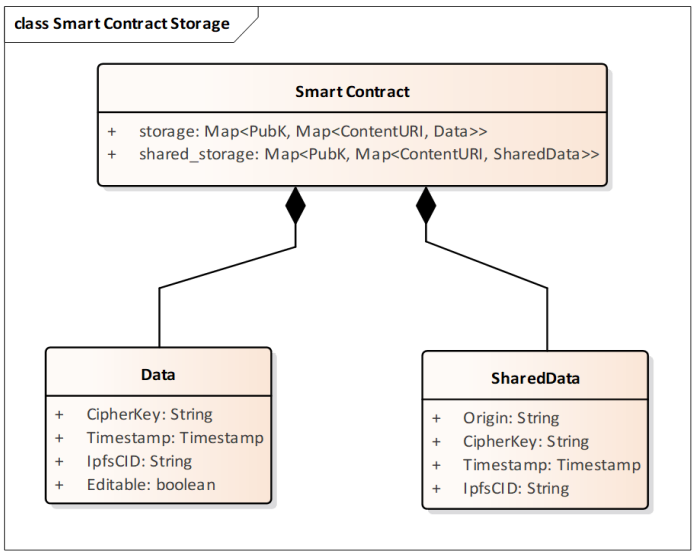


Fig. 3. Blockchain Storage Model

The memory of the Smart Contract has therefore been divided into two parts. Using Hash Tables, each User (represented by a public key) has space where he can

uniquely associate a *ContentURI* to a particular meta-data. Like the previous one, the second part is a separate storage space dedicated exclusively to the sharing of meta-data between different users and the exchange of information.

3.3 Data Storage and Management

Store Data. Once the data has been organized into the IPFS-Data structure previously described, the information is sent to the different systems in the following way, assuming that the User is already registered into the system and has its own Public (`<PubK>`) and Private Key (`<PrivK>`):

1. Before storing a data item, a User must first encrypt it with a uniformly randomly generated symmetric key `<SPrivK>`. The generation operation must be repeated for each entry. The binary data (encrypted according to a standard block cipher) is uploaded to IPFS, which returns the Content Identifier (**CID**) for addressing. At this point, the data is associated with a unique URI (*ContentURI*) generated by the User so that it is meaningful to describe the content of the data.
2. The symmetric key `<SPrivK>` is encapsulated with further encryption using its public key `<PubK>` and finally destroyed to prevent it from being recovered by an unauthorized party. The resulting cipher takes the name **CipherKey**.
3. The information is sent to Smart Contract, which takes care of storing it:
   a. **Address**: User's public key.
   b. **ContentURI**: URI that uniquely describes the data.
   c. **CipherKey**: Symmetric key, used to encrypt the data, encapsulated.
   d. **CipherAlgorithm**: Algorithm used for symmetric data encryption.
   e. **IpfsCID**: CID returned by IPFS when the file is uploaded.
   f. **Editable**: Boolean value indicating whether the CID field can be updated in the future or not.

Some data types may be subject to change over time (such as, for example, contact information), so the ability has been given to updating the CID reference when necessary. IPFS does not allow deletion of a file; obsolete versions of the data will remain present in the network as history.

No particular cipher has been specified in the framework; instead, the User can choose the best encryption algorithm for the type of data he intends to store. As is the case with hash functions on IPFS,[2] it was deemed appropriate to adopt a design that can be easy to evolve.

[2] https://multiformats.io/multihash/

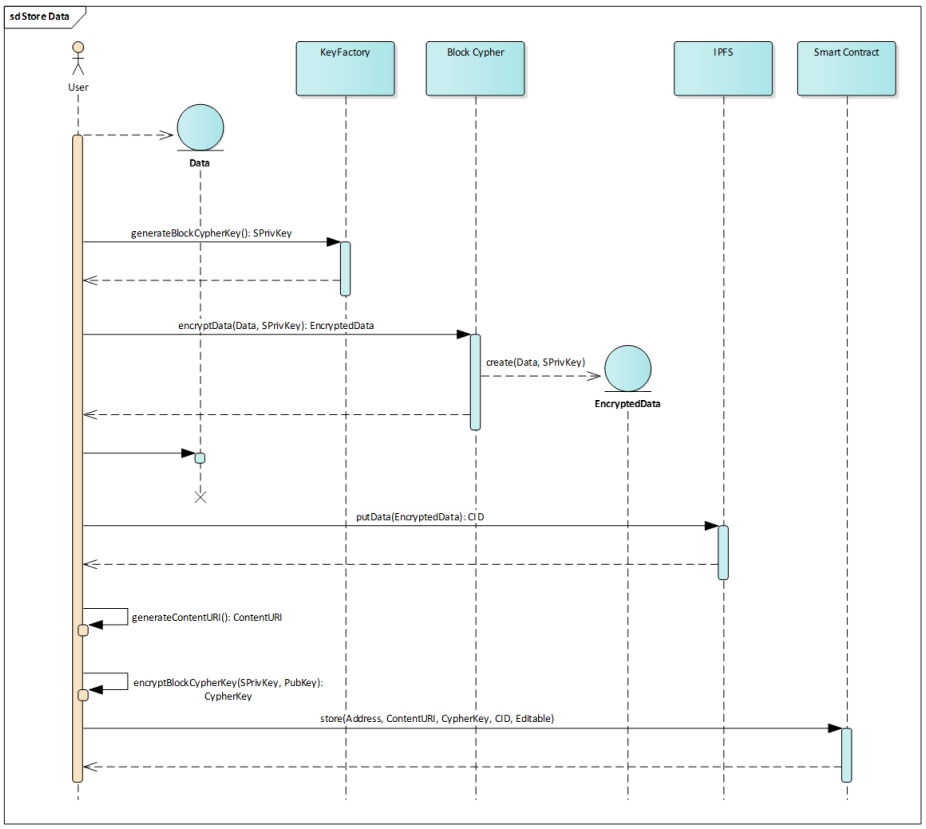


Fig. 4. Store data sequence diagram

Retrieve data. Retrieving data from decentralized services is done in a similar way:

1. User can request from the Smart Contract the tuple of information `(CypherKey, CipherAlgorithm, IpfsCID)` by giving as input the pair of values `(PubK, ContentURI)`.
2. At this point the symmetric key must be extracted from the CypherKey value, decryption is then performed using `<PrivK>` to trace back to the symmetric key `<SPrivK>`.
3. The encrypted data is downloaded from IPFS using its addressing CID, decrypted using `<SPrivK>`, which for security reasons is immediately destroyed.

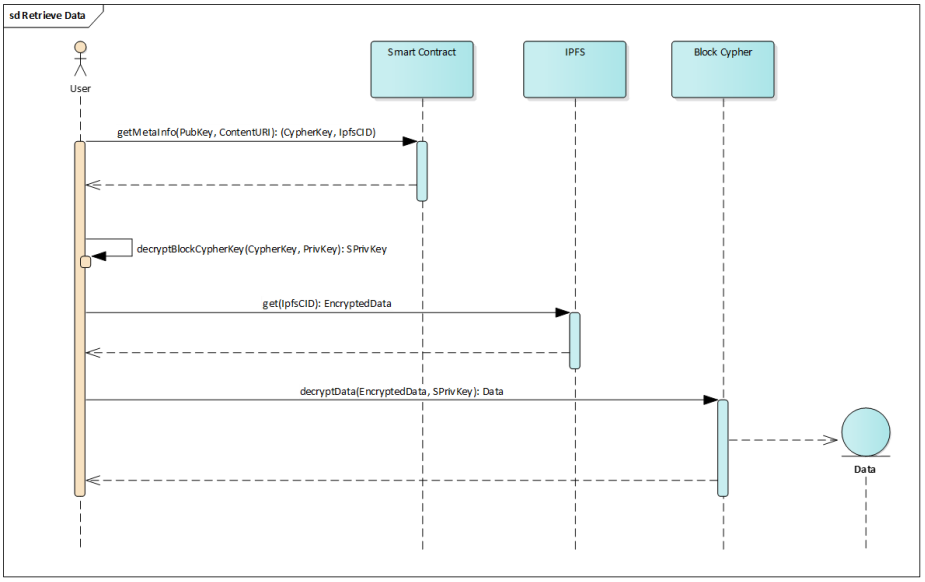


Fig. 5. Retrieve Data sequence diagram

Share Data. Sharing can only be done after the data has been uploaded to Smart Contract. Therefore, it is assumed that the User who wants to share a piece of information is already in possession of `<SPrivK>`, IpfsCID, and ContentURI, as well as the User's public key to whom he wants to view his data. Information exchange is made through the transfer of the `<SPrivK>` information between Users. No other private information are exchanged during sharing process, in particular User's private keys.

The following information is sent to the Smart Contract:

1. **SharedAddress**: Public key of the User with whom the data is to be shared.
2. **ContentURI**: URI that uniquely describes the data.
3. **CipherKey**: `<SPrivK>` used to encrypt the data, encapsulated using *SharedAddress* as the encryption key.
4. **CipherAlgorithm**: Algorithm used for data encryption.
5. **IpfsCID**: CID returned by IPFS when uploading the file.

The User to whom the files are shared will use the same steps described in the previous paragraph to access the data.

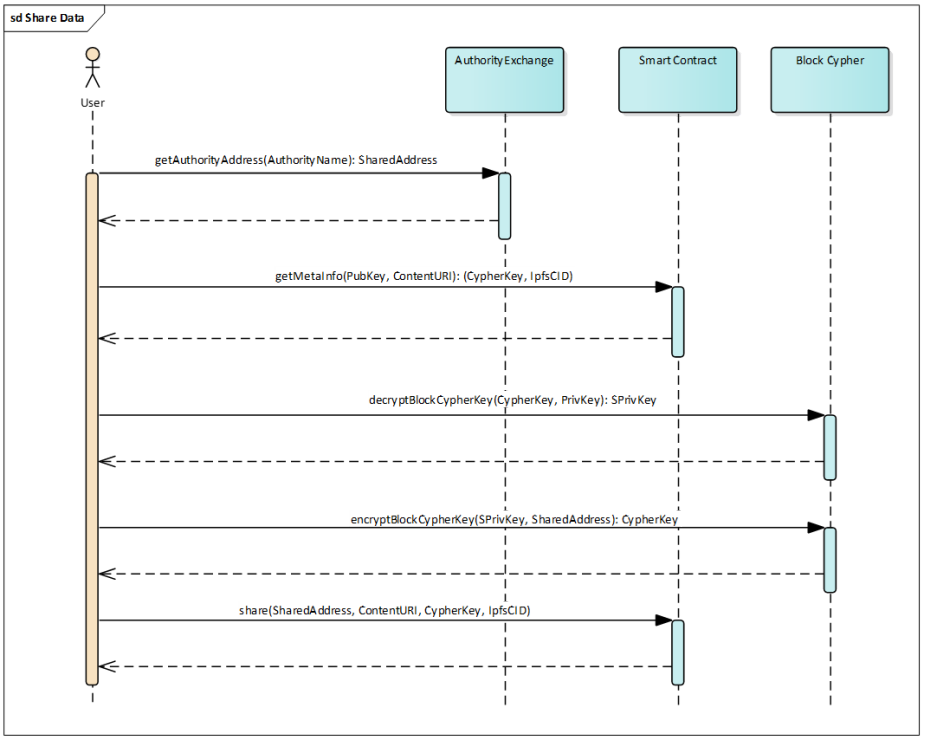


Fig. 6. Share data sequence diagram

Some fields are automatically managed by Smart Contract to ensure data security and certification:

- The *timestamp* fields are always populated with the time of the block in which the transaction is made. This field allows for certification of data entry over time.
- When sharing data, the origin field is automatically set as the User's public key making the transaction. This allows certifying the tracking of the data.

3.4 Implementation into Ethereum Virtual Machine

The system has been implemented on Ethereum Blockchain via Smart Contract in Solidity language.

Main Data Structures in the Smart Contract

```
struct Data {
    string cypherkey;
    string cypheralgorithm;
    uint256 timestamp;
```

```
        string ipfscid;
        bool editable;
        bool isValue;
    }

    struct SharedData {
        address origin;
        string cypherkey;
        string cypheralgorithm;
        uint256 timestamp;
        string ipfscid;
        bool isValue;
    }

    mapping(address => mapping(bytes32 => Data))
dataStorage;
    mapping(address => mapping(bytes32 => SharedData))
sharedDataStorage;
```

The main structures of the Smart Contract are represented by the hash tables that contain the data. A public key is uniquely associated with a second hash table to which a ContentURI can be associated with metadata.

Store Function

```
function storeData(bytes32 contentUri, string memory
cypherkey, string memory cypheralgorithm,string memory
ipfscid, bool editable) public returns(bool success) {
        //Deny storage if contentUri is already used

require(!dataStorage[msg.sender][contentUri].isValue);

        //Store values in data structure
        dataStorage[msg.sender][contentUri].cypherkey =
cypherkey;

dataStorage[msg.sender][contentUri].cypheralgorithm =
cypheralgorithm;
        dataStorage[msg.sender][contentUri].ipfscid =
ipfscid;
        dataStorage[msg.sender][contentUri].timestamp =
now;
```

```
        dataStorage[msg.sender][contentUri].editable =
editable;
        dataStorage[msg.sender][contentUri].isValue =
true;

        //Emit Blockchain event when new data is
inserted
        emit DataInsert(msg.sender, contentUri);

        return true;
    }
```

The functions within the Smart Contract have been realized, trying to perform as few operations as possible to increase efficiency and reduce costs.

Blockchain Events list

```
    event DataInsert(address authority, bytes32
contentUri);
    event DataModified(address authority, bytes32
contentUri);
    event DataShared(bytes32 contentUri, address from,
address to);
```

Each operation (Insert, Edit, Share) is associated with an event, i.e., a signal propagated throughout the peer-2-peer network, which is essential for the development of decentralized applications and communication between the various stages.

4 Conclusions

In conclusion, in this paper, we tried to investigate one of the possible applications of Blockchain and decentralized systems for data management and sharing. The results obtained agree with what was researched. Indeed, we succeeded in realizing a scalable structure for the secure collection of data to be easily shared within a well-defined network and protected from the privacy point of view. The system thus obtained is therefore independent of a central organization, collaborative and safe suitable for extraordinary situations such as, for example, a global pandemic. The encryption techniques used are safe because they follow the current standards without making changes to the algorithms. Moreover, they have been used so as not to create dependencies in the use of keys. Each cryptographic primitive uses its private key.

This work opens the possibility of multiple future developments. The first one is undoubtedly the research of new technologies on which to implement the framework.

One of the critical aspects of this system is the non-negligible cost to maintain security and decentralization of data. The information saved on IPFS must be spread on the network using FileCoin[3], while the operations on the Blockchain have a variable and often unpredictable price. The use of these public technologies may not be feasible in practice, as they are not sustainable in terms of costs. It would therefore be interesting to study alternative implementations of currently experimental technologies, such as, for example, Blockchain with integrated decentralized storage[4]. A second aspect that could be expanded would be to build a Public Key Infrastructure (PKI) to manage the public keys of all users, also using decentralized models and approaches [20] [21]. Finally, Smart Contract events can be utilized to build applications that operate on the data in real-time, such as building indexing services, notification services etc.

References

1. Chen, Chiang, and Storey, "Business Intelligence and Analytics: From Big Data to Big Impact," *MIS Quarterly*, vol. 36, no. 4, 2012, doi: 10.2307/41703503.
2. C. J. Wang, C. Y. Ng, and R. H. Brook, "Response to COVID-19 in Taiwan," *JAMA*, vol. 323, no. 14, Apr. 2020, doi: 10.1001/jama.2020.3151.
3. Q.-V. Pham, D. C. Nguyen, T. Huynh-The, W.-J. Hwang, and P. N. Pathirana, "Artificial Intelligence (AI) and Big Data for Coronavirus (COVID-19) Pandemic: A Survey on the State-of-the-Arts," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3009328.
4. I. E. Agbehadji, B. O. Awuzie, A. B. Ngowi, and R. C. Millham, "Review of Big Data Analytics, Artificial Intelligence and Nature-Inspired Computing Models towards Accurate Detection of COVID-19 Pandemic Cases and Contact Tracing," *International Journal of Environmental Research and Public Health*, vol. 17, no. 15, Jul. 2020, doi: 10.3390/ijerph17155330.
5. M. Ienca and E. Vayena, "On the responsible use of digital data to tackle the COVID-19 pandemic," *Nature Medicine*, vol. 26, no. 4, Apr. 2020, doi: 10.1038/s41591-020-0832-5.
6. G. Cantelli *et al.*, "The European Bioinformatics Institute: empowering cooperation in response to a global health crisis," *Nucleic Acids Research*, vol. 49, no. D1, Jan. 2021, doi: 10.1093/nar/gkaa1077.
7. D. Ichikawa, M. Kashiyama, and T. Ueno, "Tamper-Resistant Mobile Health Using Blockchain Technology," *JMIR mHealth and uHealth*, vol. 5, no. 7, Jul. 2017, doi: 10.2196/mhealth.7938.
8. S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of Internet content delivery systems," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, Dec. 2002, doi: 10.1145/844128.844158.
9. A. Dolgui, D. Ivanov, S. Potryasaev, B. Sokolov, M. Ivanova, and F. Werner, "Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain," *International Journal of Production Research*, vol. 58, no. 7, Apr. 2020, doi: 10.1080/00207543.2019.1627439.
10. K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare Blockchain System Using Smart Contracts for Secure Automated

[3] https://docs.filecoin.io/about-filecoin/ipfs-and-filecoin/#data-storage-incentives
[4] NeoFS: public distributed decentralized object storage: https://fs.neo.org/

Remote Patient Monitoring," *Journal of Medical Systems*, vol. 42, no. 7, Jul. 2018, doi: 10.1007/s10916-018-0982-x.
11. S. Park, G. J. Choi, and H. Ko, "Information Technology–Based Tracing Strategy in Response to COVID-19 in South Korea—Privacy Controversies," *JAMA*, vol. 323, no. 21, Jun. 2020, doi: 10.1001/jama.2020.6602.
12. A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," Aug. 2016, doi: 10.1109/OBD.2016.11.
13. K. Christodoulou, P. Christodoulou, Z. Zinonos, E. G. Carayannis, and S. A. Chatzichristofis, "Health Information Exchange with Blockchain amid Covid-19-like Pandemics," May 2020, doi: 10.1109/DCOSS49796.2020.00071.
14. H. R. Hasan *et al.*, "Blockchain-Based Solution for COVID-19 Digital Medical Passports and Immunity Certificates," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3043350.
15. S. Sharifian and R. Safavi-Naini, "Information-theoretic Key Encapsulation and its Applications." 2021.
16. K. Bhargavan *et al.*, "Formal Verification of Smart Contracts," Oct. 2016, doi: 10.1145/2993600.2993611.
17. K. K.-Y. Lee, W.-C. Tang, and K.-S. Choi, "Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage," *Computer Methods and Programs in Biomedicine*, vol. 110, no. 1, Apr. 2013, doi: 10.1016/j.cmpb.2012.10.018.
18. M. Vukolić, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," 2016.
19. W. Zou *et al.*, "Smart Contract Development: Challenges and Opportunities," *IEEE Transactions on Software Engineering*, 2019, doi: 10.1109/TSE.2019.2942301.
20. M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," 2016.
21. M. Al-Bassam, "SCPKI," Apr. 2017, doi: 10.1145/3055518.3055530.