On the Decentralization of Health Systems for Data Availability: a DLT-based Architecture

On the Decentralization of Health Systems for Data Availability: a DLT-based Architecture

Gioele Bigini*, Mirko Zichichi†, Emanuele Lattanzi*, Stefano Ferretti*, Gabriele D'Angelo‡
*Department of Pure and Applied Sciences, University of Urbino, taly
†Ontology Engineering Group, Universidad Politécnica de Madrid, pain
‡Department of Computer Science and Engineering, University of Bologna, Italy
g.bigini@campus.uniurb.it, mirko.zichichi@upm.es, {emanuele.lattanzi,stefano.ferretti}@uniurb.it, g.dangelo@unibo.it

***Abstract*—Mobile devices entered people's lives by leaps and bounds, offering various applications relying on private third-party entities to manage their users' data. Centralized control of personal health data endangers the privacy of the users directly involved. In the future, there will likely be a trend toward decentralizing the health data collection, relieving central entities of this task. This comes with several challenges in a decentralized environment, such as avoiding a single point of failure to guarantee data availability. The following work proposes an architecture based on Distributed Ledger Technology to allow users to decide on their data while ensuring availability by employing social networks. We will outline the mechanisms behind data storage and the implications of using smart contracts in the architecture. In concluding the work, we show the developed architecture and results deriving from its assessment, highlighting possible use cases applied to the specific health data management context.**

***Index Terms*—Distributed Ledger Technology, Smart Contracts, Health Data, Distributed Storage, Social Networks**

I. Introduction

Personal digital technologies are constantly evolving and are the primary source of information generation. They brought a fundamental transformation in people's lives, but the data generated usually ends up in private use for reasons related to the privacy of an individual. The shift to a decentralized paradigm now seems immediate, not only to protect the individual but, more importantly, to enable new technologies revolving around data management. Storing data in centralized data silos makes it inaccessible to the public and disconnected from other data [1], [2]. This mechanism hampers innovation above all. Based on this, interest in decentralizing data management is proliferating and with great promise to enable these conditions. The healthcare sector can benefit significantly from decentralizing information from centralized systems. This is because there are so many new implications of doing this, ranging from contributing personally to advancing new medical studies in a disintermediate way and getting new solutions on your own directly from devices that become incredibly efficient and intelligent [3], [4]. Indeed, this path is not easy, there are considerable barriers to be addressed in terms of security and privacy, but it is the most suitable vision to enable the digital health field.

This work has received funding from Regione Marche with DDPF n. 1189, the EU's Horizon 2020 research and innovation programme under the MSCA ITN grant agreement No 814177 LAST-JD-RIoE and from the University of Urbino through the "Bit4Food" research project.

Assuming we had such a decentralized approach available, we could enable a new way of exploiting data in the healthcare space by storing our data ourselves. However, new issues would be introduced, such as the data availability problem. The availability of a piece of data indicates its ability to be accessed at any time, at any place. However, decentralizing an infrastructure around individuals implies their commitment to ensuring that their data is always accessible. Therefore, guaranteeing such an approach means having to find strategies to guarantee it. We believe that social techniques and mechanisms can be vital in maintaining such an infrastructure. We mean social techniques designed to increase end-user involvement with the problem, such as the introduction of gamification techniques and social network activities. This work aims to ensure decentralized data availability by creating a circle of trusted users to avoid the single point of failure. These users' could securely share stored information and delegate their information when needed, helping to improve data availability while ensuring privacy.

In this work, we propose a decentralized architecture for health data sharing focused on solving the data availability issue through social networks. A Distributed Ledger Technology (DLT) stores universal and immutable resource identifiers for data and provides smart contracts to ensure proper access control associated with each piece of data. We provide the first attempt at a mechanism geared toward increasing data availability in a decentralized context. So, we propose an architecture based on DLTs, smart contracts, Distributed File Storage (DFS), and social networks. Their combination ensures the decentralized distribution of health data preserving data sovereignty, confidentiality, and secure access control. We provide a use case related to the Internet of Medical Things (IoMT) applications that demonstrate, through a data sharing scenario, how to leverage the proposed system to increase data availability. As a conclusion for the work, we provide the system evaluation showing that the proposed system is feasible.

The remainder paper is organized as follows. Section II presents the background, while Section III describes the related work. Section IV specifies the proposed architecture. In Section V, we describe a IoMT application use case. Performance is evaluated in terms of measured latency in Section VI to demonstrate feasibility before the conclusions, in Section VII.

II. Background

In this section, we explain the technologies used to develop the proposed architecture.

A. Distributed Ledger Technology

DLTs were born to shift trust from a human intermediary handling a transaction between two parties to a peer-to-peer (P2P) protocol that allows two or more parties to conduct transactions directly. The resistance to manipulation makes DLTs an up-and-coming technology for developing new types of applications where immutability and transparency are a requirement. Examples of such applications can be found in DLTs that implement smart contracts, for example, through Ethereum [5], [6]. However, it is essential to note that the ability to incorporate smart contracts usually involves the blockchain trilemma, which means that scalability and responsiveness are affected [7]. A smart contract is a code executed in a DLT environment or the source code from which that code was compiled [5]. This code is executed deterministically by different DLT participants, who receive the same inputs and perform a computation that leads to the same outputs. When a smart contract is deployed on the DLT and the issuer is confident that the code embodies the expected and correct behaviour (e.g., by examining the code), transactions originated by that contract do not require the presence of a third party to have value. This principle is based on the assumption that most DLT nodes are honest (i.e., the opposite of an attacker node) and follow the same protocol.

B. Decentralized File Storage

A Decentralized File Storage (DFS) offers an alternative way of storing files to traditional client-server models, i.e. a domain name is provided and then translated into an IP address [8], [9]. A DFS comprises a network of peer nodes that have their storage and follow the same protocol for storing and retrieving content. Several alternatives are available to organize the P2P system, ranginf from an opportunistic network to a structured organization [10]. In order to know which DFS node in the network owns the requested content, it is possible to rely on a distributed hash table responsible for mapping contents, i.e., files and directories, to peers that own them. In content-based addressing, content is queried directly through the P2P network rather than establishing a connection with a server. For instance, in the IPFS [8] DFS, content is retrieved using its CID, i.e., an identifier obtained through the content's hash digest. DFS follows this approach and offers increased data availability and resilience by using data replication.

C. Decentralized Data Availability

The current practice of online service providers is to store users' data in centralized cloud storage system as it is less expensive and more reliable than local storage in personal devices, thus is less prone to data loss and provides high data availability. However, using a centralized server to store published data raises the issue of data privacy and sharing for specific applications involving personal data.

On the opposite side, users can use their personal devices to store and make data available to other users, creating a P2P network. In such a distributed system, however, ensuring both data availability, persistence and dissemination is a challenge due to the dynamic and distributed nature of the system [11]. If personal data are stored directly on users' personal devices and are available only when the user's system is online, data availability is a challenge [12]. In fact, a node managed directly by a user in a decentralized scenario does not offer the same availability properties as a centralized cloud node management.

III. Related Work

With the emergence of the first proposals to use DLT-based systems beyond finance, i.e. cryptocurrencies, some researchers have already found a relationship between DLT and personal data management. In this context, the general approach is to securely store access control policies on DLTs so that the applicant can be made aware of his or her permissions to access his or her personal data stored outside the DLT [6], [13]. However, few attempts have been made in the past to ensure data availability and most of the works focuses on centralized cloud technologies. Li and Dabek [14] argue that, when implementing a distributed storage infrastructure in P2P systems, a node should choose its trusted node neighbours, i.e., the nodes with which it shares resources, based on existing social relationships, rather than randomly, e.g., their friends and colleagues. Gracia-Tinedo et al. [15] showed that pure friend-to-friend storage systems have poor Quality-of-Service (QoS), mainly due to availability correlations, and proposed a hybrid architecture to combine it with cloud storage services. Liu et al. [16] present a decentralized online social network designed to manage data without compromising user privacy, i.e., user's data are replicated to trusted servers controlled by friends.

However, today's work primarily focuses on designing intelligent data replication and storage policies. The approach proposed by Koll et al. [17] exchanges recommendations among socially related nodes to efficiently distribute replicas of a user's data among suitable nodes carefully selected in the OSN. In the approach developed by Olteanu et al. [18], preferences are given to nodes when it comes to selecting nodes for storing data (and their replicas) published by a user. The user's online friends have the highest priority. When all friends are offline, data is stored in nodes not part of the user's circle of friends. Guidi et al. [12] use the Interplanetary File System (IPFS) to build a decentralized because of its decentralized nature for DOSN. In their work, they inspect whether IPFS is a good choice as data storage for Decentralized Social Applications.

In this work, we differ from the works in the state of the art because we want to give more space to how a mixed DLT and DFS context can potentially be exploited to ensure greater data availability in a decentralized context. We focus on enabling users to replicate data, decide over it and involve them in

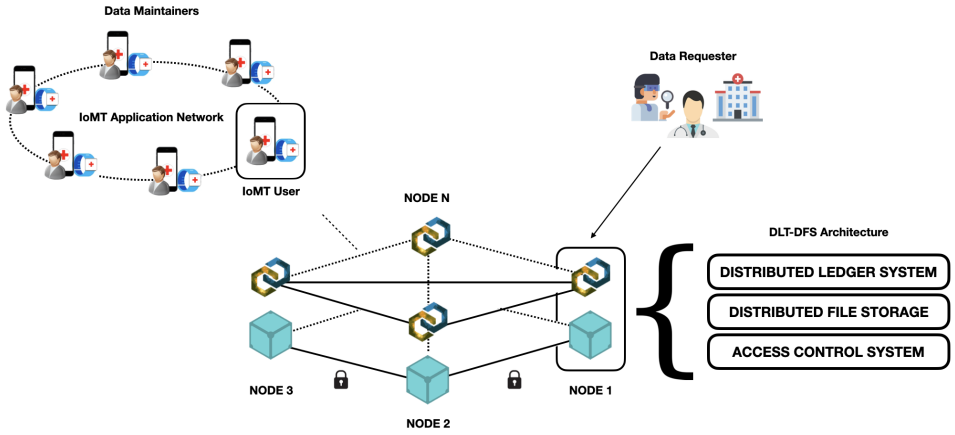


Fig. 1. Decentralization Health Data Architecture for Data Availability.

storage and policy decisions in advance by employing a social network in a decentralized scenario.

IV. Proposed Architecture

In this section, we describe our proposed system. The goal of this system is to provide a decentralized architecture to involve users in the decisions made concerning their data. A mechanism based on social networks consisting of a voting system has the dual purpose of representing an access mechanism to data and increasing data availability. Users maintain their data, store it in their nodes, and create social networks to make joint decisions about the data, allowing the users in the network to replicate it and eventually delegate it. Through this mechanism, users of IoMT devices can directly own their personal data while ensuring availability through policies, i.e. delegation. A specific example for which the network could be relevant is in the event that a user becomes incapable, for some reason, of making decisions, such as in the case of an accident or sudden and unexpected disability. Still, he might have delegated in advance his rights to trusted individuals in the social network. In this paper, we will not dwell on the possible policies to identify delegates, or on devising proper multi-party decision making schemes, since it is closely related to the specific use case. Indeed, our work is more focused on the provision of a decentralized architecture fostering this kind of healthcase applications. Thus, in the rest of the paper we will consider a naive data authorization scheme based on a voting system, i.e. the data owner and his delegates can vote to decide if a requester can get access to the data.

We describe the system architecture with the aid of Figure 1:

- **IoMT Application and Social Network** - end-users interact with an application for managing health data and providing social features for data availability.
- **Distributed Ledger Technology** - the DLT, through smart contracts, are used to reach consensus on one's data, enabling secure access, processing and sharing of medical data among different e-health entities.
- **Decentralized File Storage** - the DFS is responsible for facilitating data sharing and providing secure storage of information.
- **Access Control System** - the authorization mechanism coupled with approaches close to social networks, enables the decentralization of users' health data and increase availability.

This architecture was designed with a set of principles, functional and non-functional requirements in mind: (i) *Data Validation*: the integrity of data generated by (or on behalf) of users must be guaranteed and verified. To this end, the system takes full advantage of the untamperability property of DLTs. (ii) *Traceability*: not only the integrity of personal data, but also their life cycles must be guaranteed and verified. Also in this case, the system takes advantage of DLTs and their smart contract features. (iii) *Privacy-by-Design*: while we need to make it difficult to change or delete data from the ledger, at the same time, if we intend to comply with regulations, e.g., the General Data Protection Regulation (GDPR) [19]. In the following, we will describe each architectural component.

A. IoMT Application and Social Network

The IoMT, which enables remote monitoring, screening, and treatment of patients through telehealth, has been successfully adopted by caregivers, health care providers, and patients. IoMT-based smart devices and their applications are having a dizzying impact ubiquitously, particularly in the global pandemic state. The introduction of social networks could increase the possibility of ensuring the greater availability of data. In the architecture, three main actors are identified that create such a social network:

- **IoMT User**: an IoMT user collects data through an IoMT device. An example of data used is health data, such as data related to one's postural condition. Then, through the IoMT application, the user creates a personal social network by adding other users whom we call data maintainers. IoMT users manage their node of the system described in this section and use the IoMT application to interact with the underlying components, i.e., to store data

in the DFS and to manage the access policies through the DLT.

- **Data Requester**: on the other hand, the data requester is generally a professional, researcher, or any entity that needs to take advantage of the data granted by the IoMT user and that needs to acquire permissions. Section IV-D explains the mechanism for handling requests.
- **Data Maintainer**: To keep track of requests and to allow access to data, the IoMT user relies on its social network, i.e., a network of data maintainers who are none other than other IoMT users running a node. Based on the IoMT user policy, data can be exchanged or delegated to the data maintainers in such a way data availability is ensured. It is not necessary for a IoMT user to worry about their node being online constantly because requests can be fulfilled by others in their trusted social network.

B. DLT

At the core of the architecture, the DLT layer provides a network of peer nodes, i.e. data maintainers, operating a smart contract enabled permissioned ledger, which makes it possible to store the entire history of data transactions and, consequently, the requests made to the access mechanism. This layer ensures immutability and transparency with respect to the data records and enables the execution of the smart contracts needed for the access control system (Section IV-D). Every time an operation is carried out on the smart contract, it is reflected to all the maintainers in the network that keep the distributed ledger integrity. A correct setup of a permissioned network eases the compliance with regulations such as the GDPR, as opposed to a public permissionless DLTs [20].

C. DFS

A DFS is used to store health data in an encrypted form and to replicate data in the network of data maintainers (that also run the DLT network). To maintain data validation, integrity and traceability, once a piece of data is published in the DFS, i.e., IPFS, the returned CID (i.e., hash pointer) is asynchronously referenced into the DLT. Data protection is maintained due to the fact that all data is encrypted at the User Interface/IoMT application level, and the DLT only stores hash pointers. DLTs are designed to make it difficult to tamper and to be transparent.Therefore, one approach to meet the Privacy-by-Design requirements is to implement off-chain storage of personal data and only store hash pointers on-chain [20]. This solution has the additional benefit of improving performances and providing higher availability for data reads and writes without introducing central trusted parties [19].

D. Access Control System

Smart contracts can be used to involve IoMT users in data management. In fact, a smart contract can enable data maintainers to accept or reject a result based on pre-established rules. This is fundamental to provide data availability and the possibility to continue the authorization service even when the IoMT user (the data owner) is offline. Each data maintainer, including the data owner, can vote through a smart contract whether or not to give data access authorization to a data requester. The idea is to implement a smart contract that provides a list of lists representing the social network constituted by the data maintainers and the list of their votes. Its functionality is described below:

- **Create()**: this operation is dedicated to create a new data request. This request is addressed to only one data maintainer, which then notifies through the DLT that a new request has been received and makes an entry in the ledger so that all the other maintainers can cast a vote.
- **Accept()/Reject()**: this operation allows voting on the ballot. The operation allows the ledger to be updated with the vote of the considered maintainer. Each maintainer has its own identity within the ledger and can write its vote within the smart contract. Based on the policy chosen for vote validation, a minimum time is required before closing the vote.
- **Get()**: this operation allows retrieving information saved on the ledger. This can be done by requesting a specific voting identifier or by recalling all votes.

V. IoMT Applications Use Case

In conventional healthcare environments, health data are collected through personal mobile devices and generally stored in centralized locations. IoMT devices are thus forced to pre-process data on board or to hideinformation. The majority of health data are then hardly accessible or take the form of open datasets, of little use to interested stakeholders. Because of this, the traditional healthcare data management infrastructure is mostly self-managed or outsourced to third-party experts. In this context, it is therefore difficult to make the best use of the information collected by IoMT devices and avoid raising additional privacy, security and infrastructure cost issues [21].

Recently, however, DLT-based systems are proposing an overhaul of architectures by applying a different philosophy to data management, potentially including any data, such as those in the health care domain. We propose a use case that falls into this category. Our architecture provides a decentralized sharing of health data, ensuring that data can be transacted between institutions and individuals by storing provenance and immutability. However, these architectures being able to be fully decentralized, suffer from the problem of data availability, i.e. they may not guarantee stakeholders continuous access because the user providing data suffers from a disability or his node is offline. For this reason, we referred to an approach involving a social network that constitutes a network of trust and enables the user with the potential of delegation.

A. Embedding Social Networks in the Internet of Medical Things Scenario

We consider a scenario where an IoMT user, Alice, collects her data through her smartphone. We refer to her as the data owner. Another system user is her physiotherapist Bob, i.e., the data requester. Alice does not trust large institutions and prefers keeping her data on her smartphone or sharing it with

her trusted network of individuals. But, her smartphone is not always online or could be lost, so the services using the policies she defined could not always work. To address the issue, Alice trusts her family and allows them to replicate it. This way, she avoids being a single point of failure through her household's and family members' devices, which we call data maintainers. Through the creation of a network of trust, the system is fault-tolerant in the event of the shutdown of one of the nodes, and data and policies are available most of the time. In this work, we consider Alice using a platform collecting health data related to her postural stability. With the platform she is storing sensitive personal data along with the results of the measurements she performs. Her data comprises two main parts: general personal information and medical health records. Examples of personal information include age, gender, and weight, while medical health records depend on the topic, i.e. medications and treatments.

The IoMT application allows the user to see the users added to their network and who participates in maintaining the data by contributing to its availability, i.e., data maintainers. The application also allows users to interact with their system node, thus enabling them to send or retrieve data from the DFS, be updated on external requests, vote on the smart contract in the DLT, and receive information on network participants. The full sequence for a standard access control process within her social network involves the following ten steps:

1) An incoming request from a data requester is forwarded to a data maintainer node.
2) The data maintainers create the record of the request, allowing all participants to vote.
3) The DLT replies with an acknowledgement.
4) The IoMT Users express their vote through their IoMT applications that are registered into the DLT.
5) The DLT replies with an acknowledgement.
6) After voting, the data maintainer checks if other maintainers have expressed their vote by following the specific policy related to the request.
7) A positive outcome grants permissions to the requester user along with requested data.
8) A negative outcome results in a denied access.

VI. System Performance

In the following sections, we describe the experimental environment and evaluate the implemented architecture.

A. Experimental Environment

The system on which we performed the experiments is a network of four nodes based on Hyperledger Fabric. Nodes reside in different continents: Europe, America and Asia. The machines have 2 cores, with 4GB RAM, 50GB of storage and Ubuntu 18.04 LTS as the operating system. Since Fabric relies on a specific service (called Orderer) for proper operation, the fourth node is used for ledger transaction ordering purposes.

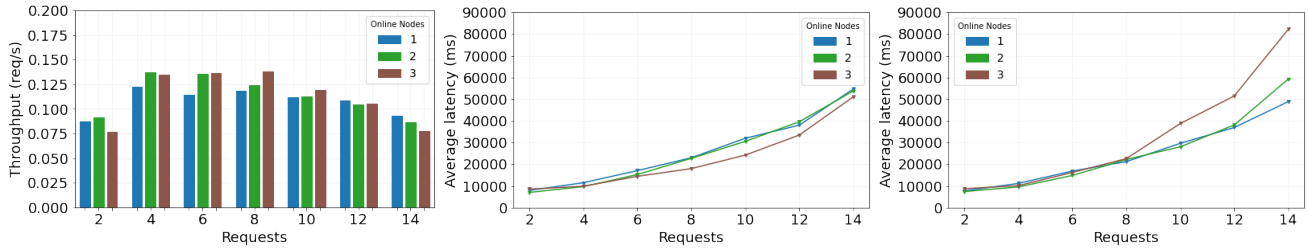
B. Testing the Distributed Ledger System

The experiment was carried out to verify the architecture's performance in case the data maintainers failed (i.e., unable to serve requests). The study was conducted following steps 1 through 8 described in Section V-A and consisted of three operations that interact with the smart contract: (i) Create(); (ii) Accept()/Reject(); and (iii) Get(). During the test, taking into account the delay of the real data maintainer in reacting to a new vote request with a parameter given randomly by a Poisson process with an average λ = 1000ms. We collected information on the following parameters and metrics:

- Fixed parameters: the maximum number of active data maintainers n was set to 3. For each test, the same queries were repeated five times. This means that we averaged the times of the same tests.
- Independent parameters: the active data maintainers t of the scheme (t, n) varies in the tests from 1 to 3, representing the increased availability of working nodes in the network. A second parameter is the number of requests per second generated by requesting users, which varies from 2 to 14.
- Element-dependent metrics: request latency, i.e., the time between sending the request and its actual completion.

C. Results

Figure 2 shows the system throughput, write and update operations as requests per second and the number of online maintainer nodes increase. An increase in the number of maintainer nodes in the network corresponds to higher data availability, as they are reachable. If only one maintainer node is online, then all requests are redirected to it. The results show a clear dependence on the number of requests per second and the value of t. Plot (a) shows the throughput of the system as the number of requests per second increases. The throughput is lower in the case where the nodes are not all active, and a peak performance increase is evident when we are in presence of about 8 requests to the data maintainers. This is verified before the threshold mentioned. After that threshold, the overall performance deteriorates and the throughput flip and gets worst globally and with more nodes involved. The chart provides a measure of scalability, meaning that the system is less efficient as the number of requests per second increases. Nevertheless, the results obtained are reasonable considering the conditions: the system remains resilient to failures and can always respond to requests even under stress. Another aspect to consider is how the throughput slowly gets worst and flips between different thresholds. This is a consequence of the increase in concurrent maintainers updating the same data and the number of requests to resolve, which causes concurrency issues that slightly affect the final performance. Regarding the Write and Update operations (Plots b and c), it is interesting to highlight how they keep slowly deteriorating. That is, we expect that we can be more efficient at maximum availability. In contrast, the rightmost chart related to the Update operation in Plot (c) shows an apparent worsening trend in the condition. The



(a) System Throughput (b) Writing Records into the DLT (c) Updating Records into the DLT

Fig. 2. DLT throughput and requests latencies.

explanation for this is what was already mentioned before. It demonstrates that simultaneous update interaction of multiple data maintainers (higher value of t) causes longer wait times on the ledger, most likely related to ledger access conflict management. In the best case, the DLT should establish about 8 simultaneous concurrent network connections per node, as performance can be assumed to degrade beyond this number. This ensures latencies of about less than 4 seconds on average. By increasing the requests, we fall into the worst case where the average latency could double.

VII. Conclusion

To date, data management systems in health-care are mainly centralized, and in most cases, data sharing is based on agreements, limiting innovation in digital health. In the case of IoMT, this factor weighs even more heavily, because these devices are extremely widespread and generate a huge amount of data that could be used for beneficial purposes. In this paper, we introduced a DLT-based access mechanism that could be used to provide more robust security while preserving data protection and introducing the idea of creating trusted social networks to manage user data and enable data sharing. The system can then ensure that patients have complete control over access to their medical records, securely stored on DFS, that only verified participants can interact with sensitive patient data, and that they can be protected during arrangements made on a DLT. Experimental evaluation of the overall architecture shows that failures in the network still provide the ability to reach data across DLT and DFS networks with increased availability. As future work, we plan to deploy our solution in a network with a larger number of data maintainers to test its scalability further and concretely reflect on decentralized social applications that can also involve gamified contexts.

References

[1] W. Christl, K. Kopp, and P. U. Riechert, "How companies use personal data against people," tech. rep., Cracked Labs, 2017.

[2] M. Zichichi, S. Ferretti, and G. D'Angelo, "A distributed ledger based infrastructure for smart transportation system and social good," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, pp. 1–6, 2020.

[3] R. De Michele and M. Furini, "Iot healthcare: Benefits, issues and challenges," in *Proceedings of the 5th EAI international conference on smart objects and technologies for social good*, pp. 160–164, 2019.

[4] G. Bigini, V. Freschi, and E. Lattanzi, "A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision," *Future Internet*, vol. 12, no. 12, p. 208, 2020.

[5] P. De Filippi, C. Wray, and G. Sileno, "Smart contracts," *Internet Policy Review*, vol. 10, no. 2, 2021.

[6] M. Zichichi, S. Ferretti, and G. D'Angelo, "A framework based on distributed ledger technologies for data management and services in intelligent transportation systems," *IEEE Access*, 2020.

[7] J. Sedlmeir, P. Ross, A. Luckow, J. Lockl, D. Miehle, and G. Fridgen, "The dlps: a new framework for benchmarking blockchains," 2021.

[8] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[9] M. Zichichi, S. Ferretti, and G. D'Angelo, "On the efficiency of decentralized file storage for personal information management systems," in *Proc. of the 2nd International Workshop on Social (Media) Sensing, co-located with 25th IEEE Symposium on Computers and Communications 2020 (ISCC2020)*, pp. 1–6, IEEE, 2020.

[10] S. Ferretti, "Shaping opportunistic networks," *Computer Communications*, vol. 36, no. 5, pp. 481–503, 2013.

[11] S. Ferretti, "Gossiping for resource discovering: An analysis based on complex network theory," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1631 – 1644, 2013. Including Special sections: High Performance Computing in the Cloud and Resource Discovery Mechanisms for {P2P} Systems.

[12] B. Guidi, A. Michienzi, and L. Ricci, "Data persistence in decentralized social applications: The ipfs approach," in *Consumer Communications & Networking Conference (CCNC)*, pp. 1–4, IEEE, 2021.

[13] G. Zyskind, O. Nathan, *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, pp. 180–184, IEEE, 2015.

[14] J. Li and F. Dabek, "F2f: Reliable storage in open networks.," in *IPTPS*, 2006.

[15] R. Gracia-Tinedo, M. S'nchez-Artigas, and P. Garcia-Lopez, "F2box: Cloudifying f2f storage systems with high availability correlation," in *2012 IEEE Fifth International Conference on Cloud Computing*, pp. 123–130, IEEE, 2012.

[16] D. Liu, A. Shakimov, R. Cáceres, A. Varshavsky, and L. P. Cox, "Confidant: Protecting osn data without locking it up," in *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pp. 61–80, Springer, 2011.

[17] D. Koll, J. Li, and X. Fu, "Soup: an online social network by the people, for the people," in *Proceedings of the 15th International Middleware Conference*, pp. 193–204, 2014.

[18] A. Olteanu and G. Pierre, "Towards robust and scalable peer-to-peer social networks," in *Proceedings of the Fifth Workshop on Social Network Systems*, pp. 1–6, 2012.

[19] M. Zichichi, S. Ferretti, G. D'Angelo, and V. Rodríguez-Doncel, "Data governance through a multi-dlt architecture in view of the gdpr," *Cluster Computing*, pp. 1–32, 2022.

[20] M. Finck and F. Pallas, "They who must not be identified—distinguishing personal from non-personal data under the GDPR," *International Data Privacy Law*, vol. 10, pp. 11–36, 03 2020.

[21] J. Indumathi and al., "Block chain based internet of medical things for uninterrupted, ubiquitous, user-friendly, unflappable, unblemished, unlimited health care services (bc iomt u 6 hcs)," *IEEE Access*, vol. 8, pp. 216856–216872, 2020.