Decentralized Health Data Management: An IPFS-based Approach and Performance Evaluation

Decentralized Health Data Management: An IPFS-based Approach and Performance Evaluation

Francesco Franco*, Alessandro Bogliolo*, Sara Montagna*, Luca Bedogni†, Stefano Ferretti‡
*Department of Pure and Applied Sciences (DiSPeA)
University of Urbino, Italy - {name.surname}@uniurb.it
†Department of Physical Sciences, Informatics and Mathematics (FIM)
University of Modena and Reggio Emilia, Italy - {name.surname}@unimore.it
‡Department of Computer Science and Engineering (DISI)
University of Bologna, Italy - {name.surname}@unibo.it

***Abstract*—Current health data management relies on centralized architectures that create a single point of failure, limit patient autonomy, and increase vulnerability to data breaches and vendor lock-in. This paper presents a decentralized approach to continuous health monitoring through the integration of wearable devices and distributed file systems. We implemented an Android application that collects physiological data and contextual information from a wearable device, storing it on the IPFS via Pinata API. Additionally, we propose a blockchain architecture for role-based access control. Performance evaluation comparing our IPFS-based implementation against Firebase Real-Time database reveals that the resource requirements remain negligible for modern smartphones while achieving significant benefits, including no single point of failure, enhanced data portability, and patient data sovereignty. The results demonstrate that decentralized health data management is technically feasible on mobile devices, offering an alternative approach to traditional centralized health data architectures.**

***Index Terms*—Distributed File System, Health data management, Mobile Health Application**

I. Introduction

In recent years, the proliferation of wearable devices such as smartwatches and fitness bands has led to a new paradigm of continuous, patient-centric care. Instead of sporadic clinical assessments, these devices allow continuous monitoring of vital signs such as heart rate, blood pressure, and oxygen saturation. These real-time data streams allow for early anomaly detection and personalized interventions and foster proactive user engagement in health management. However, data management remains a critical challenge. Most current solutions rely on proprietary mobile apps that collect sensor data and store it in vendor-controlled, centralized systems. This model tends to limit transparency, as the inner workings of data storage and processing remain opaque to the end users. Studies have shown that users are increasingly concerned about how their data is used, stored, and potentially monetized [3]. Moreover, centralized health data management, though convenient for developers and data scientists, inherently introduces vulnerabilities: it creates a single point of failure (SPOF), reduces user control over sensitive information, and leads to the formation of so-called data silos —isolated repositories that hinder interoperability and make it difficult for patients to port their data across different platforms and services. This siloed structure not only limits data portability but also prevents the emergence of integrated health ecosystems. As highlighted in [7], healthcare data breaches are more frequent than in other sectors, due to the higher black market value of medical records. Beyond concerns over security and privacy, centralized architectures reduce patient autonomy and compromise long-term resilience. To address these issues, decentralized approaches have been proposed that couple the use of Distributed File Systems (DFS) together with smart contracts executed on blockchain. DFS presents a compelling alternative since, unlike monolithic server-based systems, it distributes data across a decentralized network of peer nodes. Through content-addressing cryptographic hashing, every record obtains a unique, tamper-evident identifier. Replication across multiple peers eliminates single points of failure while users retain control over access policies by choosing which nodes host or "pin" their data. To favor traceability and access control policies, data are encrypted, and their references are stored and managed thanks to dedicated smart contracts [12].

In this work, we present the design, implementation, and evaluation of a native Android application that integrates continuous health data collection with DFS-based storage and incorporates blockchain technology to ensure data integrity, security and governance. Specifically, the app collects heart rate, blood pressure, and contextual data from a Samsung wearable device, in our case a Galaxy Watch 5, stores it on the Interplanetary File System (IPFS), and records the corresponding Solidity-based smart contract on an Ethereum blockchain. We also evaluate the system's resource impact on the smartphone to assess feasibility for long-term use. By demonstrating that a DFS-backed architecture can operate efficiently within the constraints of consumer mobile devices, we aim to take a significant step toward more resilient, privacy-preserving health data ecosystems that empower users rather than tethering them to centralized silos.

The remainder of this paper is organized as follows: Section II outlines some related works, Section III describe the system architecture and its implementation, Section IV presents the experimental evaluation, focusing on resource consumption by

the app. Finally, Section V provides some concluding remarks.

II. Related Work

Wearable technologies have been studied for continuous health monitoring and remote patient support [6], [9], especially for the management of elderly and chronic patients [10], but the challenge of ensuring scalable, resilient, and privacy-preserving data storage remains a critical concern. The use of wearable devices with distributed storage architectures is a solution to manage and secure sensitive health-related data.

DFS, particularly the IPFS, have emerged as promising alternative to traditional centralized silos that may not completely assure privacy of information. IPFS [4] is a protocol that builds a distributed file system over a peer-to-peer network. It offers several advantages: decentralization mitigates the risk of SPOF, content replication enhances resilience and availability, and its content-addressable design ensures data integrity through cryptographic hashing. Another difference from centralized systems is the user control and autonomy from third-party providers [5]. In contrast, centralized systems, while often more efficient in terms of latency and management simplicity, are vulnerable to service outages, data breaches, and potential misuse of personal data.

To further boost these distributed storage solutions and address issues of trust, immutability, and verifiable data provenance, the integration of blockchain technology presents a convincing next step documented in the literature [1], [8]. For instance, [11] proposes a healthcare data management framework utilizing Hyperledger Fabric to enhance privacy and interoperability in healthcare data sharing. Their approach implements a permissioned blockchain network, ensuring controlled access to sensitive healthcare information through smart contracts, allowing different permissions based on the actor and purpose of the access. The solution combines blockchain with off-chain cloud storage, only hash values of medical data are stored on chain, ensuring traceability and immutability while maintaining storage efficiency. Patients maintain control of their data, supervising usage through immutable logs recorded on the blockchain.

However, blockchain systems are often characterized by slow transaction speeds and high computational requirements [2]. In particular, the distributed nature of blockchain can introduce additional network latency and increase transmission delays, especially when handling large volumes of data or a high number of users. Storing blockchain ledgers and managing complex consensus mechanisms necessitate significant storage and processing resources. For IoT devices, this resource intensity is a particular concern, as they often cannot absorb additional computing or energy requirements, necessitating optimized blockchain integration.

Accordingly, in this paper, we propose an architecture that collects health data from a wearable device, stores it on IPFS, and integrates blockchain technology to ensure data integrity and security. However, considering these limitations of blockchain systems, especially critical in IoT environments, our evaluation specifically focuses on measuring battery consumption and network traffic at the personal device level.

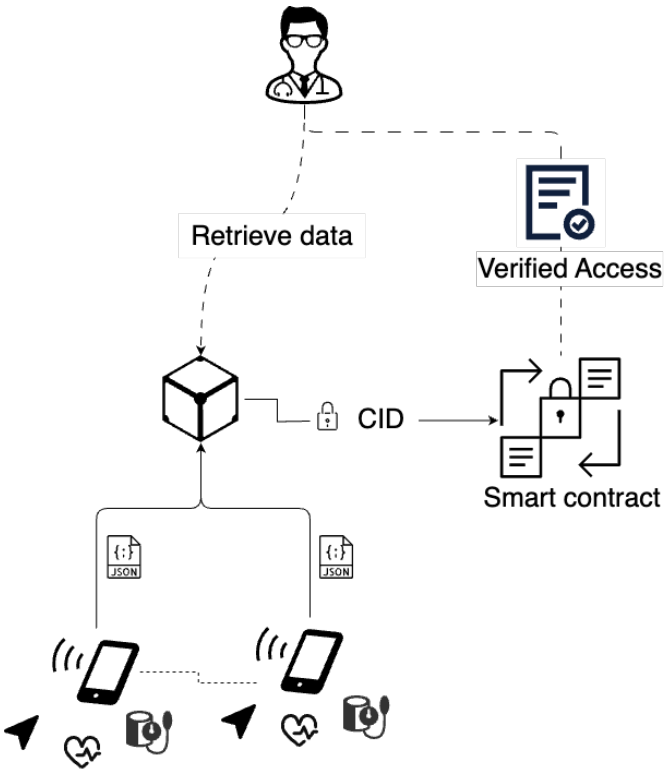

Fig. 1. Overall system architecture for decentralized health data management. Wearable devices collect physiological data processed by an Android application using Health Connect API. Data is stored locally and periodically transmitted as JSON payloads to IPFS via Pinata. Blockchain smart contracts register encrypted content identifiers to ensure data integrity and privacy, enabling healthcare providers to securely access and verify patient data while maintaining sovereignty.

III. Proposed architecture

This architecture addresses key challenges in current health data management systems by combining patient data sovereignty with clinical accessibility. In this work, we propose an architecture for collecting, storing, and decentralizing data acquired from wearable devices and users' smartphone sensors, such as location services. The system is centered around an Android application that gathers physiological metrics (e.g., heart rate and blood pressure) and contextual data (e.g., location) from wearable and smartphone sensors. To ensure standardized access to health data, the application leverages Health Connect (HC), an Android API that serves as a local hub, allowing fitness and health apps to share and store structured data on the device without requiring custom integration. Data in HC is organized into a standardized schema covering categories such as activity, vitals, nutrition, sleep, and body measurements. Physiological data such as heart rate, step count, and movement are continuously gathered by the wearable and stored in HC. Blood pressure detection requires human intervention as it must be detected by correctly positioning the arm and the sensor. It also requires

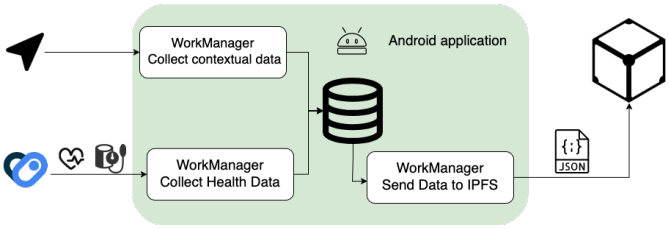

Fig. 2. Architecture of the Android application, illustrating data flow from wearable and smartphone sensors to local storage and decentralized backend (IPFS).

calibration; for instance, the Samsung smartwatch we used must be paired with a traditional blood pressure monitor to ensure accurate readings. HC also enforces fine-grained access control, requiring explicit permission to be granted for each type of data accessed by the application.

Building on this foundation, the app acts as an orchestrator of the entire data lifecycle, from acquisition and local persistence to decentralized transmission. Through HC, the application retrieves heart rate and blood pressure data made available by the wearable device. Collected data is first stored in a local SQLite database. We use the Room persistence library to provide an abstraction layer over SQLite to enable more robust schema management and safe queries. Data acquisition and transmission are handled in the background using the WorkManager API, which supports deferred, persistent jobs that can run even when the app is idle or after a device reboot. In our implementation, we configured three periodic workers using WorkManager. Each worker is scheduled to run every 30 minutes, an interval we identified as optimal by balancing several factors: it is frequent enough to capture significant variations in physiological parameters, yet not so frequent as to excessively impact battery life and network usage. This interval also ensures data freshness while maintaining system reliability under varying connectivity conditions. The workers have the following responsibilities:

- The contextual data worker retrieves the user's most recent known location. This allows blood pressure readings to be enriched with environmental context;
- The data collection worker interfaces with HC to gather physiological data such as heart rate and blood pressure, storing it locally using Room;
- The data transmission worker serializes and sends the stored health data enriched with contextual data, see Listing 1, to IPFS via the Pinata API.

The WorkManager API provides built-in retry mechanisms and fault tolerance, ensuring data collection and transmission continue reliably even when network connectivity is intermittent or device resources are constrained. This mechanism guarantees data redundancy and availability while removing reliance on centralized infrastructure. This setup, once the necessary permissions have been granted, allows the application to operate autonomously in the background with minimal user intervention. To ensure decentralization and persistent storage, the system relies on Pinata, a pinning service that interfaces with IPFS and guarantees retrievability through content-addressable identifiers (CIDs). In our implementation, to send data to IPFS, we define a Retrofit Interface that includes several endpoints for interacting with the Pinata and IPFS APIs. The data is serialized into a JSON object, see 1, and submitted to Pinata. Each payload includes both raw physiological readings and associated metadata such as timestamps, location, and heart rate aggregates.

Listing 1. Example of a JSON payload submitted to Pinata, containing a blood pressure reading, contextual location, and associated heart rate statistics.

```
{
  "id": Integer,
  "uid": String,
  "timestamp": Long,
  "timezone": Integer,
  "systolic": Double,
  "diastolic": Double,
  "description": String,
  "bodyPosition": String,
  "latitude": Double,
  "longitude": Double,
  "heartRate": {
    "hrStart": Long,
    "hrEnd": Long,
    "hrAVG": Double,
    "hrMIN": Double,
    "hrMAX": Double,
    "hrMC": Integer
  }
}
```

To enhance security and establish an immutable record of health data, our proposed architecture incorporates blockchain technology. When data is uploaded to IPFS, the resulting CID is encrypted and then recorded on a blockchain using a smart contract. Smart contracts are written in Solidity and deployed on an Ethereum blockchain. The smart contract associates each IPFS hash with a pseudonymized patient identifier, creating a verifiable chain for health data while maintaining privacy through pseudonymization. This approach ensures that data integrity can be verified without exposing sensitive health information on the blockchain itself. The encryption of CIDs adds an additional layer of privacy protection, ensuring that even blockchain participants cannot directly access the stored health data without proper authorization. The blockchain component adds critical capabilities essential for health data management:

- Each data upload is permanently recorded with precise timing;
- The cryptographic signatures ensure authentication of data source, establishing non-repudiation;
- The encrypted hash stored on-chain can be compared with the recalculated hashes of retrieved data, verifying data integrity while maintaining access control;
- Smart contract logic enables role-based access control, allowing operations based on clinical responsibilities.

For healthcare providers, a front-end application allows authorized clinicians and caregivers to access patient data securely. This interface implements the following workflow:

- Authenticating using role-based access control aligned with clinical responsibilities;
- Retrieval of relevant IPFS hashes from the blockchain for a specified patient;
- Verification of data integrity by comparing on-chain hashes with the retrieved data's calculated hash;
- Presentation of physiological data with contextual information to support clinical decision-making.

This architecture allows the system to operate reliably and autonomously, combining native Android APIs for health and background processing with decentralized technologies. The use of multiple workers enables fine-grained separation of concerns between data collection, contextual enrichment, and transmission. The combination of local data persistence, decentralized storage, and blockchain verification creates a robust foundation for patient-controlled health data management. In the next section, we evaluate the performance of the Android application in terms of battery consumption and network usage.

IV. Performance evaluation

To evaluate the performance impact of our decentralized architecture, we implemented two identical Android applications differing only in their data transmission backend:

- App A: Data transmission to IPFS via Pinata API, our proposed architecture;
- App B: Data transmission to Firebase Real-Time Database.

Both applications utilize the same HC integration, WorkManager scheduling, Room database for local storage, and identical JSON payload structures. This comparison allows us to isolate the performance differences attributable specifically to the choice of storage backend. The evaluation was conducted on Galaxy A54 running Android 14 over a 60-hour monitoring period. To simulate realistic usage patterns, we generated blood pressure measurements at random intervals during each day, with each measurement accompanied by contextual location data and aggregated heart rate statistics from the preceding 30-minute window. This irregular measurement schedule aims to replicate the natural variability of health monitoring in real-world scenarios, where users take measurements at different times based on their daily activities and health needs. The experimental design ensures that both applications operate under identical conditions, with the same health data samples, network connectivity characteristics, and background processing schedules. Our analysis focused on two critical performance metrics that directly impact user experience and device usability: battery consumption and network traffic efficiency. The experimental results reveal a clear tradeoff between decentralized benefits and resource consumption. As illustrated in Figure 3, the IPFS-based implementation demonstrates approximately 10.5% higher total battery consumption compared to the Firebase implementation, consuming 11.77 mAh versus 10.65 mAh over the monitoring period. When examining hourly consumption rates (Figure 4), the decentralized approach shows a 10.3% increase in power draw with 0.193 mAh/h compared to Firebase's 0.175 mAh/h. Network traffic reveals more substantial differences. Figure 5 demonstrates that the IPFS implementations transmit approximately 97% more data than the Firebase counterpart, with 438.75 KB versus 222.52 KB total network traffic. The observed performance differences stem from fundamental architectural distinctions between centralized and decentralized systems. The 10.5% battery increase results from several factors: HTTP-based API communication with Pinata requires additional connection establishment and authentication overhead compared to Firebase's optimized mobile SDK, more intensive JSON processing for metadata responses, and extended network radio usage for larger data transfers. However, the energy requirements remain minimal for both scenarios, less than 12 mAh over a 60-hour monitoring period, and represent a negligible impact on modern smartphone battery life. The 97% increase in network traffic reflects the architectural differences between database-optimized SDKs and file storage APIs. The additional overhead primarily results from Pinata API requirements, including HTTP authentication headers and comprehensive response metadata (CID, upload confirmations, file information, and timestamps) that increase payload size compared to Firebase's response. The performance implications extend beyond simple resource consumption metrics and must be evaluated within the healthcare context. The 10.5% battery increase and 97% network traffic increase represent the quantifiable cost of achieving complete data sovereignty through distributed file systems. From a healthcare perspective, these results are particularly noteworthy when considering the privacy landscape. The moderate energy overhead becomes acceptable when weighted against the elimination of data breach risks, vendor lock-in, and potential service discontinuation of the centralized systems. While Pinata API integration provides broader compatibility and easier development, it introduces communication overhead that specialized systems like Firebase can avoid. The experimental results demonstrate that decentralized health data transmission represents a practically viable alternative to centralized approaches, with an acceptable performance overhead relative to the privacy and sovereignty benefits achieved.

V. Conclusion

The paper presents a decentralized approach to health data management through the integration of wearable devices, DFS, and blockchain technology. Our implemented solution demonstrates that decentralized health data collection and storage are feasible on mobile devices. The performance of the mobile application reveals that DFS introduces moderate but acceptable overhead compared to centralized alternatives. The increase in battery consumption and network traffic represents a reasonable tradeoff for achieving data sovereignty. The experimental results demonstrate several key advantages of

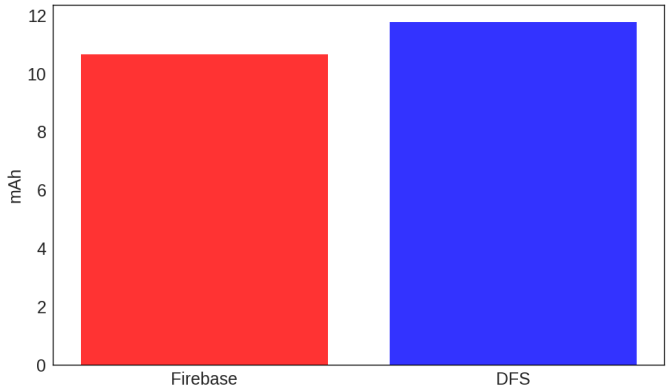

Fig. 3. Total battery consumption comparison between Firebase (10.65 mAh) and IPFS-based (11.77 mAh) implementations over the monitoring period, showing a 10.5% increase for the decentralized approach.

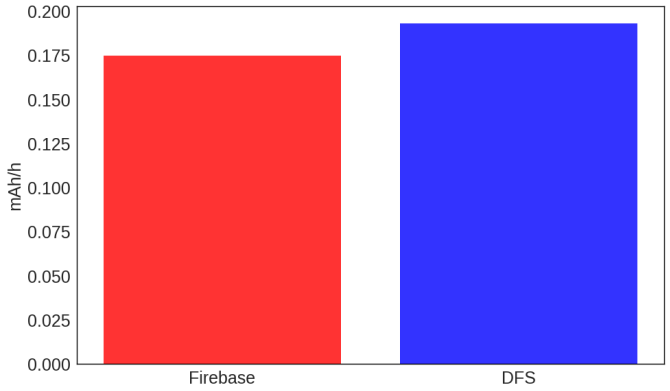

Fig. 4. Hourly battery consumption rates demonstrating the power efficiency difference between Firebase (0.175 mAh/h) and IPFS-based (0.193 mAh/h) data transmission, representing a 10.3% increase in the decentralized implementation.

the implemented mobile architecture: elimination of single point of failure through distributed storage, enhanced data portability and interoperability via standardized APIs, and reduced vulnerability to vendor lock-in and service discontinuation. The proposed blockchain integration would provide additional benefits, including role-based access control and privacy protection through encrypted CID while maintaining patient pseudonymity. Future research includes implementing the proposed blockchain architecture to validate its practical feasibility and performance impact. Additionally, enhancing the WorkManager implementation with intelligent data prioritization could significantly optimize network usage and battery consumption by performing preliminary data analysis on the smartphone and immediately transmitting only critical values that require urgent clinical attention while maintaining a single daily backup for routine measurements. This approach would reduce unnecessary network traffic while ensuring the timely delivery of potentially life-critical information.

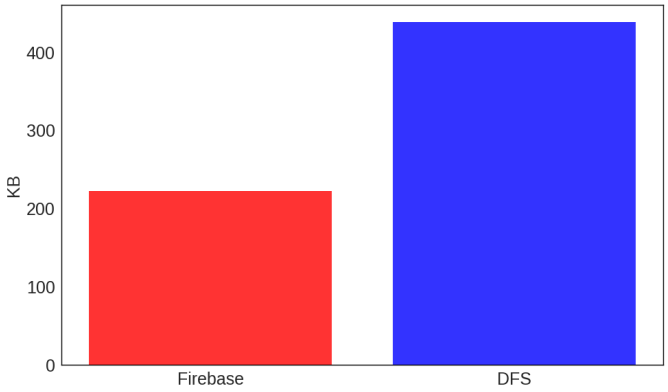

Fig. 5. Network traffic comparison showing total data transmitted by Firebase (222.52 KB) versus IPFS (438.75 KB) implementation, illustrating the 97% increase in network utilization required for decentralized operation.

Acknowledgment

Funding: This work has been partially funded by the European Union - NextGenerationEU within the framework of PNRR Mission 4 - Component 2 - Investment 1.1 under the Italian Ministry of University and Research (MUR) programme "PRIN 2022" - grant number 2022N2NH42 - SmartShires - CUP: H53D23003570006

References

[1] Asad Abbas, Roobaea Alroobaea, Moez Krichen, Saeed Rubaiee, S. Vimal, and Fahad M. Almansour. Blockchain-assisted secured data management framework for health information analysis based on internet of medical things. *Personal and Ubiquitous Computing*, 28(1):59–72, 2024.

[2] Turki Ali Alghamdi, Rabiya Khalid, and Nadeem Javaid. A survey of blockchain based systems: Scalability issues and solutions, applications and future challenges. *IEEE Access*, 12:79626–79651, 2024.

[3] Nasser Alhammad, Mohannad Alajlani, Alaa Abd-Alrazaq, Gregory Epiphaniou, and Theodoros Arvanitis. Patients' perspectives on the data confidentiality, privacy, and security of mhealth apps: systematic review. *Journal of Medical Internet Research*, 26:e50715, 2024.

[4] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[5] Christian Esposito, Alfredo De Santis, Genny Tortora, Henry Chang, and Kim-Kwang Raymond Choo. Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE cloud computing*, 5(1):31–37, 2018.

[6] SM Riazul Islam, Daehan Kwak, MD Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. The internet of things for health care: a comprehensive survey. *IEEE access*, 3:678–708, 2015.

[7] HIPAA Journal. Healthcare data breach statistics, 2025. Accessed: 2025-05-05.

[8] Md. Moniruzzaman, Seyednima Khezr, Abdulsalam Yassine, and Rachid Benlamri. Blockchain for smart homes: Review of current trends and research challenges. *Computers & Electrical Engineering*, 83:106585, 2020.

[9] Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2009.

[10] Anu Raj and Shiva Prakash. A privacy-preserving authentic healthcare monitoring system using blockchain. *International Journal of Software Science and Computational Intelligence*, 14(1):1–23, October 2022.

[11] Qianyu Wang and Shaowen Qin. A hyperledger fabric-based system framework for healthcare data management. *Applied Sciences*, 11(24), 2021.

[12] Mirko Zichichi, Stefano Ferretti, Gabriele D'Angelo, and Víctor Rodríguez-Doncel. Data governance through a multi-dlt architecture in view of the gdpr. *Cluster Computing*, 25(6):4515–4542, 2022.