Journal of the American Medical Informatics Association, 2025, **32(2)**, 296–307
https://doi.org/10.1093/jamia/ocae288
Advance access publication 21 November 2024
Research and Applications

Research and Applications

Distributed, immutable, and transparent biomedical limited data set request management on multi-capacity network

Yufei Yu, BS[1,2,†], Maxim Edelson, MS[3,†], Anh Pham, PhD[1,2,†], Jonathan E. Pekar, PhD[4], Brian Johnson, BS[1,2], Kai Post, MS[1,2], Tsung-Ting Kuo, PhD[1,5,6,*]

[1]Division of Biomedical Informatics, Department of Medicine, University of California San Diego, La Jolla, CA, 92093, United States, [2]Department of Biomedical Informatics, University of California San Diego Health, La Jolla, CA, 92093, United States, [3]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, 92093, United States, [4]Institute of Ecology and Evolution, University of Edinburgh, Edinburgh, EH9 3FL, United Kingdom, [5]Department of Biomedical Informatics and Data Science, School of Medicine, Yale University, New Haven, CT, 06510, United States, [6]Department of Surgery, School of Medicine, Yale University, New Haven, CT, 06520, United States

*Corresponding author: Tsung-Ting Kuo, PhD, Department of Biomedical Informatics and Data Science and of Surgery, 100 College Street, School of Medicine, Yale University, 100 College Street, New Haven, CT, 06510, United States (tsung-ting.kuo@yale.edu)

†Y. Yu, M. Edelson, and A. Pham contributed equally to this work.

Abstract

Objective: Our study aimed to expedite data sharing requests of Limited Data Sets (LDS) through the development of a streamlined platform that allows distributed, immutable management of network activities, provides transparent and intuitive auditing of data access history, and systematically evaluated it on a multi-capacity network setting for meaningful efficiency metrics.

Materials and Methods: We developed a blockchain-based system with six types of smart contracts to automate the LDS sharing process among major stakeholders. Our workflow included metadata initialization, access-request processing, and audit-log querying. We evaluated our system using synthetic data on three machines with varying specifications to emulate real-world scenarios. The data employed included ∼1000 researcher requests and ∼360 000 log queries.

Results: On average, it took ∼2.5 s to register and respond to a researcher access request. The average runtime for an audit-log query with non-empty output was ∼3 ms. The runtime metrics at each institution showed general trends affiliated with their computational capacity.

Discussion: Our system can reduce the LDS sharing request time from potentially hours to seconds, while enhancing data access transparency in a multi-institutional setting. There were variations in performance across sites that could be attributed to differences in hardware specifications. The performance gains became marginal beyond certain hardware thresholds, pointing to the influence of external factors such as network speeds.

Conclusion: Our blockchain-based system can potentially accelerate clinical research by strengthening the data access process, expediting access and delivery of data links, increasing transparency with clear audit trails, and reinforcing trust in medical data management. Our smart contracts are available at: https://github.com/graceyufei/LDS-Request-Management.

Key words: data sharing; blockchain; privacy and security; data protection.

Introduction

The cross-institutional data requesting and data sharing landscape

In the modern healthcare landscape, clinics and institutions routinely generate vast datasets throughout patients' courses of care.[1,2] As these datasets include direct human biometrics, they serve as a rich resource for researchers, especially when studies require insights into the timing and location of health events. This leads to the use of the Limited Data Set (LDS),[3] a category of datasets that includes up to 2 types of these Protected Health Information[4] or Personal Identifiable Information,[5] along with other non-identifiable data. Such constraints on LDS enable researchers to study specific populations without majorly compromising individual privacy.[6] Proper management of LDS access contributes to safeguarding privacy accordant with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rules[7] and reinforcing public trust in healthcare practices, thereby empowering the robust growth of data-driven biomedical research. The LDS sharing protocol often encompasses proof of signed data use agreements (DUAs),[8] as well as user demonstration of valid data-access credentials, such as possession of relevant biomedical training certificates from the Collaborative Institutional Training Initiative (CITI) program.[9] As mandated by HIPAA, a DUA defines authorized users, delineates permissible actions, restrictions, and exemptions, includes legal provisions against re-identifying or directly contacting data subjects, and may also describe breach-reporting protocol.[10] Meanwhile, training certificates ensure that researchers are well-versed in the necessary regulatory

Received: August 13, 2024; **Revised:** October 14, 2024; **Editorial Decision:** November 3, 2024; **Accepted:** November 5, 2024

standards governing the specific dataset.[9] Given the need of researchers requesting data from organizations under different regulatory regimes,[11,12] establishing an effective, institution-spanning process to verify these credentials is crucial to the continuity and efficiency of biomedical and clinical research.

Manual data access processes

Nonetheless, as depicted in Figure 1A, the current data access and delivery process is hindered by manual procedures involving various stakeholders, from researchers who seek data, to the data concierge offices responsible for verifying access credentials and facilitating dataset distribution. After gathering information on the access requirements, researchers must first obtain the necessary compliance training certificates and sign the DUA specific to their desired dataset. The documents should then be sent to the concierge office who manually verifies their validity and expiration. The verification of these prerequisites at the data concierge office can often be a time-consuming task, especially when the information provided is incorrect or incomplete. This will not only prolong the researcher's wait time but also introduce variability in the verification pace due to human factors, thereby adding non-trivial time costs to the overall research timeline from days to weeks. From the perspective of data concierge staff, the human labor required scales linearly with the number of inquiries and the variety of documents related to each dataset. Moreover, this process must be repeated each time the researcher wishes to access another dataset, even if the required credentials such as CITI training certificates have been previously submitted and are still valid.

Centralized solution for data request management

To address these shortcomings and enhance the overall efficiency of data sharing, it is essential to adopt an automated system to confirm access credentials[13] and consequently approve or deny access requests. A potential and intuitive solution is the conventional model of a centralized database, which allows all relevant information of datasets, compliance certificates, and signed DUAs to be stored for future matching, thus preventing repeated submissions and verifications. However, such centralization may pose innate workflow and security challenges: (1) by design, a centralized system is susceptible to the Single-Point-of-Failure (SPoF) risk, where information becomes inaccessible during server downtime (Figure 1B).[14–16] (2) Centralization is also vulnerable to unauthorized access or alteration under compromised "admin" privileges, a less common yet detrimental event that may remain undetected for a long time.[15,17,18] (Figure 1C) (3) In an architecture shared among multiple institutions, this is even more challenging as it would require additional efforts to establish transparency with central administrators, so that auditors can independently confirm access history for auditing purposes (Figure 1D).[19]

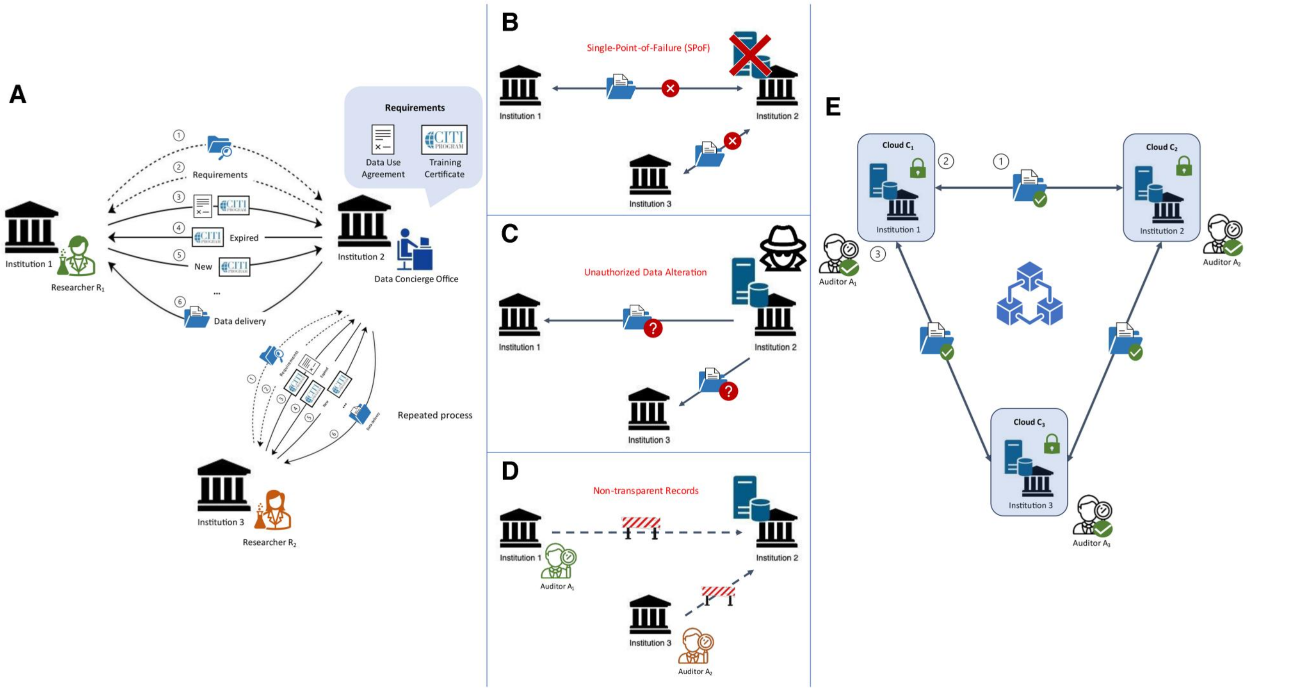

Figure 1. Comparison of limited dataset request process between three institutions under different systems. (A) Traditional email process. *Step 1*. Researcher R_1 from Institution 1 requests a dataset from Institution 2's data concierge office. *Step 2*. Office informs of requirements: data use agreement (DUA) and training certificates. Steps 1 and 2 are skipped if requirements are online (eg, on an official website). *Step 3*. Researcher R_1 submits DUA and certificates. *Step 4*. Office notes expired certificates. *Step 5*. Researcher R_1 obtains and resubmits new certificates. *Step 6*. Office approves and delivers the dataset. Process repeats for subsequent researchers. (B) Centralized system. In this example, Institution 2 uses an automated system and serves as the centralized server; upon fulfilling dataset requirements, data is instantly sent to researchers. However, Institution 2 can present a single-point failure leading to data inaccessibility during server downtime. (C) Another centralized server risk of unnoticed, unauthorized data changes by administrators, resulting in different datasets for identical requests. (D) The third centralized server difficulty for auditors at other institutions to access request history without database permissions since the audit logs are not transparent to all parties. (E) Decentralized blockchain system. Each institution has a cloud environment with database storage, offering multiple benefits including: (1) no single-point failure, ensuring constant data accessibility; (2) immutable data and request records on the blockchain, preventing unauthorized changes; and (3) transparent cross-institutional request history queries for auditors.

Needs for a decentralized data requesting system

Given inherent issues of the centralized database, a decentralized system that can function without a centralized repository emerges as the more technically fit solution to prevent: (1) central server outages, (2) data mutability, and (3) process opacity. In particular, the decentralized ledger blockchain has been advocated for use in various healthcare domains[20] such as clinical data management,[21–23] clinical research data logging,[24] secured and unified health records access,[25] genomic data access and gene-drug interaction recording,[26,27] health data compliance and abuse detection,[28] medical image sharing,[29] and credential exchange for data sharing among verified healthcare entities.[30] Compared to the limitations of a centralized server, blockchain is known for (1) robustness against SPoF; information is broadcast across the network in a peer-to-peer fashion, enabling continuation of interaction even if a network participant (node) ceases to function, thereby eliminating the SPoF risk inherent in a centrally-coordinated schema. Furthermore, blockchain enhances security through (2) its resistance to unauthorized data modifications with the intentional duplication of data at each node, ensuring easy detection of record modification. (3) Lastly, blockchain improves widespread transparency in data sharing; as each node holds a copy of the entire data ledger, all information becomes accessible and verifiable by any participant.[15,31] Another advantage of blockchain infrastructure is the integration of smart contracts,[31] which are customizable programs stored and executed on the blockchain that automate credential verification and data distribution. Smart contracts inherit blockchain's core features: decentralization, immutability, and transparency for the *code* (in addition to the *data*). These characteristics of blockchain and smart contracts make them particularly well-suited to enhance automation, accuracy, and decentralization in LDS sharing. Figure 1E summarizes the benefits of a decentralized blockchain system.

Related work

Several blockchain- and smart contract-based proposals have aimed to refine the data request workflow, each with varied focuses and degrees of complexity.[32–36] For example, a past study demonstrated blockchain's capability to store and retrieve training certificates, but it has yet to address the automated verification of access credentials and the release of access links upon approval.[35] Another study targeted storing dataset metadata in the approval protocol and negotiating DUA terms multilaterally, but has yet to incorporate the credential verification process.[32] Additionally, a different approach used blockchain as an honest service to facilitate access based on user trust and reputation, granting or denying access based on predicted noncompliance risk, and has yet to verify regulatory compliance through formal training certificates and/or DUA.[34] Regarding the traceability of data request history, several studies leveraged blockchain's transparency for "built-in" auditing, displaying all data requesting activities. However, the recorded information was presented as either identifiers[33] or hashed strings,[36] which requires further maneuvers to concretely locate actors and actions within a specific timeframe; a domain-specific design that allows intuitive filtering and auditing has yet to be developed. From an architectural standpoint, previous proposals focused on single-site blockchain evaluations[34] or multi-site blockchain implementations where each site had identical hardware specifications.[33] One study used computers with varying capacities without conclusively determining the impact of this design on system performance.[36] Additionally, while several studies have measured system efficiency,[32] more statistically comprehensive evaluations have yet been conducted. Thus, there is a need for a non-homogeneous architecture that accommodates configuration heterogeneity and is systematically evaluated for time efficiency, providing a statistically significant metric.

In summary, there has yet to be a data-sharing request management system that can: (1) automatically handle key aspects of the LDS sharing mechanism, including verification of access credentials and delivery of data links; (2) allow users to query access activities with transparent ease; and (3) provide comprehensive evaluations in cross-site environments with varied computational configurations.

Objective

We aimed to expedite the sharing request of LDS through (1) the development of a streamlined platform that may allow distributed, immutable management of network activities, (2) provide transparent and intuitive auditing of data access history, and (3) systematically evaluate it on a multi-capacity network setting for meaningful efficiency metrics.

Methods

Method overview

A schematic representation of our proposed workflow is in Figure 2, with a sample network of three institutions, each deployed within a cloud environment with varying computational settings. From researchers' standpoint, they can request datasets from any institution within the system, and they can expect to receive fast responses. Specifically, successful requests yield data sharing links, while unsuccessful ones result in the reasons of denial (eg, absence of a certificate, expired certificate, or lack of a signed DUA). Instead of directly storing the entire protected content of LDS datasets on-chain, our system employs data sharing links to ensure data privacy. Additionally, auditors from any institution have the capability to query the data request history across the network (ie, auditor at Institution 3 can query requests not only of Institution 3 but also of Institution 1 and Institution 2). The following sections provide detailed specifics of our system: section "Smart contract architecture" outlines the smart contract architecture; section "Workflows" discusses system workflows; section "Data" details our test data; section "Implementation" describes system implementation; and section "Evaluation settings" covers evaluation setting.

Smart contract architecture

Our framework includes 2 categories of smart contracts: *stakeholders* and *utility*. In particular, the *stakeholder* contracts include Institution, Researcher Management, and Data Concierge, to represent each critical entity involved in the data access request process. These smart contracts can manage, monitor, and record metadata according to their specific roles. Next, *utility* contracts consist of Log, Connector, and Date Time, which act as supportive utility components within the system. The main architecture governing the interplay of these smart contracts is in Figure 3. Further details of each smart contract are provided in Table 1.

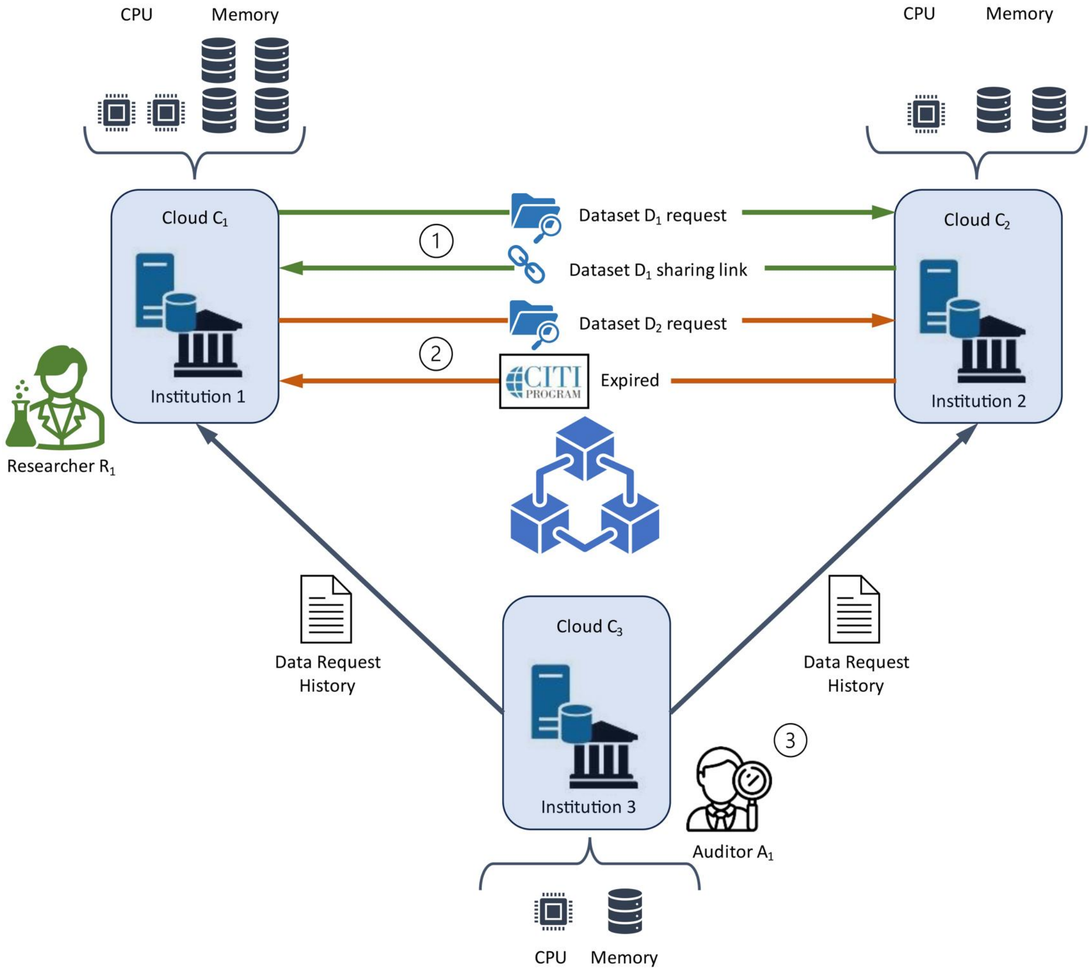

Figure 2. Overview of the method workflow in a blockchain network comprising three institutions, each with a different computational configuration. (1) Successful Data Request. Researcher R_1 requests dataset D1 from Institution 2 and promptly receives a data sharing link. (2) Unsuccessful Data Request. Researcher R1 from Institution 1 requests dataset D2 from Institution 2 but is immediately notified that their certificates have expired. (3) Auditing Process. An auditor from Institution 3 who wishes to look up the requesting history of datasets from Institution 1 and 2 can query other institutions' history seamlessly.

Workflows

Our system streamlines LDS access requests with three subworkflows:

1) *Metadata initialization*. Building on the capabilities demonstrated in previous studies where the metadata and PDF files of credential certificates can be stored on-chain, our initialization step extends the data request pipeline by transmitting metadata regarding the dataset into our system through the invocation of the Data Concierge smart contracts (Table 2a). All submitted dataset metadata are coupled with their respective DUA. Records of researchers who have signed the DUAs are maintained in the Data Concierge smart contract. Similarly, the Researcher Management contracts are invoked to transmit metadata of certificate types and expiration dates (Table 2b). Each certificate is individually imported, mirroring the practice of a researcher individually submitting their newly acquired certificate onto the system.
2) *Researcher access request processing*. Detailed backend interactions involved in the access request and delivery process between any two institutions are shown in Figure 4. First, a researcher initiates an access request through their institution's Researcher Management smart contract, which then forward the request to the destination institution's Data Concierge smart contract. The Data Concierge smart contract will automatically verify the validity of training certificates and DUA metadata. Upon successful verification, a data sharing link is provided to the requester. In cases of denial, the system provides specific reasons for the rejection. Additionally, the Data Concierge smart contract would help record every request, along with its outcome and timestamp, to the Log contract, with which auditors can subsequently query. Descriptions of the access request fields are in Table 2.
3) *Audit-log querying*. Auditors from any institution can retrieve the request history via the log query function of the Log smart contract. For example, an auditor can find records of instances where *Researcher A* asked for *Dataset D* from *Institution 1* during the period between *Time Point M* and *N*. The details of the audit-log query are described in Table 2d.

Data

To evaluate our prototype, we generated synthetic data designed to emulate real-life scenarios of researchers

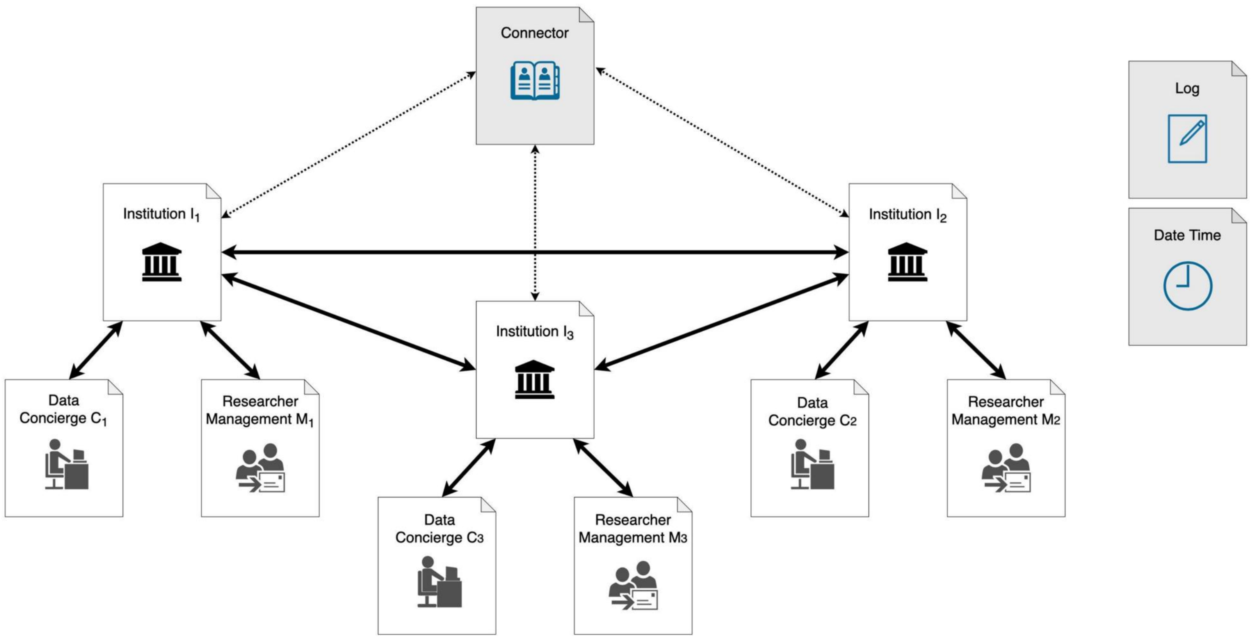

Figure 3. Smart contract architecture of three institutions. The system incorporates six types of smart contracts. Connector, Log, and Date Time function as utility smart contracts within this architecture, highlighted in gray. Each Institution smart contract supervises a Data Concierge and a Researcher Management contract unit. Institution smart contracts communicate with one another via the utility Connector. Further details on the roles and interactions of these smart contracts are provided in Section "Smart contract architecture".

Table 1. Detailed description of individual smart contracts.

Category	Smart contract	Description
Stakeholder	*Institution*	• Manage other types of stakeholder contract • Communicate with Connector and other Institution contracts
	Researcher Management	• Store and update researcher's certificate metadata (email, completed certifications, etc.) • Let researchers submit data access requests
	Data Concierge	• Store and update metadata of datasets and their corresponding DUA • Receive and process access requests
Utility	*Log*	• Store timestamped records of data access requests • Allow querying of all records
	Connector	• Store and pass along addresses of each member contract
	Date Time	• Convert between calendar and Unix time units

requesting data across three institutions; each institution consisted of 20 researchers and 10 datasets, accompanied by the corresponding DUA per dataset. Each dataset may require recipients to possess up to three distinct types of training certificates, and correspondingly, each researcher may hold up to three specific training certificates. Researcher access requests were random matched among all researchers and datasets, regardless of their institutional affiliations. With regards to the auditing functionality, we generated an exhaustive list of all possible query combinations to facilitate thorough testing. An audit-log query includes 6 inputs: Researcher Email, Institution of Dataset, Dataset, Start Time, End Time, and Response Type (details described in Table 2d). Additionally, "*" is a "wild card" parameter in audit-log queries representing all possible inputs for that parameter. We examined every possible combination of log queries, including those that produced results and those that would return an empty value. The numbers of the generated data are shown in Table 3.

Implementation

Based on prior surveys,[37,38] we chose Ethereum, an open-source blockchain platform with smart contract support,[39] to be the underlying infrastructure for our prototype due to its proven versatility and robust support in biomedical research.[20] Compared to newer alternatives like Hyperledger Fabric[40] or Corda,[41] Ethereum allows flexible configuration of both public and private networks. It also offers pseudo-anonymization through hashes, enhancing user privacy protection and facilitating regulatory compliance.[42] Additionally, Ethereum's long-established, active developer community ensures consistent, robust support across many applications.[43] We selected the Proof-of-Authority (PoA) consensus protocol,[44] designed for private blockchains where only authorized parties may join the network. The choice of private blockchain aligns with our goal of modeling a multi-node system in an environment with semi-trusted relationships among participants, unlike public blockchains which would expose data to all, including unwarranted parties.[45]

Table 2. Details of (a) dataset metadata, (b) researcher's certificate metadata, (c) researcher access request, and (d) audit-log query. "*" can be an input for any of the audit-log query fields.

Type	Field	Description	Example(s)
(a) Dataset metadata	Dataset name	Dataset's full name, which is the unique identifier of the dataset	Data Set 21
	Dataset link	Dataset's data delivery link	https://dataset_repository.com/data_set_21
	Owner researcher	Dataset owner's email	janedoe@inst2.edu
	Required certificate	List of required certificates for dataset access	Biomedical Data Only Research, Biomedical Informatics Research
	DUA number	Unique identifier links to the relevant DUA	DUADS21
	Signed researchers	List of researchers who had signed the DUA	tskuo@inst1.edu, johndoe@inst3.edu
(b) Researcher's certificate Metadata	Researcher name	Researcher's full name	Tsung-Ting Kuo
	Researcher email	Researcher's email address, which is the unique identifier of the researcher	tskuo@inst1.edu
	Certificate name	Certificate's full program name	Biomedical Data Only Research
	Certificate expiration date	Certificate's expiration date	12/19/2026
(c) Researcher access request	Researcher email	Email address of the researcher who performed the request	tskuo@inst1.edu
	Institution of dataset	Data-generating healthcare site	Institution 2
	Dataset	Full name of the requested dataset	Data Set 21
	Request time	Unix timestamp of the request time	1703126751 (December 20, 2023, 18:45:51 PST)
(d) Audit-log query	Researcher email	Email address of the researcher who performed the request	tskuo@inst1.edu
	Institution of dataset	Data-generating healthcare site	Institution 2
	Dataset	Full name of the requested dataset	Data Set 21
	Start time	Unix timestamp of query start time	1483257600 (January 01, 2017, 00:00:00 PST)
	End time	Unix timestamp of query end time	1704009600 (December 31, 2023, 00:00:00 PST)
	response type	Type of response of the requests	N (denied request)

For the development of the smart contract, we employed Solidity version 0.8.4,[46] Go Ethereum (Geth) version 1.12.7.[47] and used Remix to design and test the smart contract codes.[48] Furthermore, Web3j version 4.5.0,[49] a Java library for Ethereum smart contract interactions, was leveraged for our off-chain Java programming along with Bash scripts. Summary of the implementation is illustrated in Figure 5.

Evaluation settings

To test the system's feasibility under real-world scenarios when each site may come with varying computational capacity, we evaluated the system on three Amazon Web Service (AWS) Virtual Machines (VMs) with different hardware specifications. The system's performance may be influenced not only by the inherent speed threshold of the blockchain, which establishes a fixed upper limit,[15] but also by the specifications of the system's hardware and network capacities, including factors such as CPU and RAM. The configurations of the VMs were as follows: the large VM (ie, Institution 1) had 4 vCPUs, 16 GB of RAM, and 200 GB of storage; the medium VM (ie, Institution 2) had 2 vCPUs, 8 GB of RAM, and 100 GB of storage; and the small VM (ie, Institution 3) had 2 vCPUs, 4 GB of RAM, and 50 GB of storage. Our performance evaluation metric was time-based, a concrete metric most relevant to clinicians and researchers and focused on several key operations: smart contract deployment, metadata initialization (including researcher's certificate metadata and dataset metadata, as described in Table 2a and Table 2b, respectively), researcher access request processing (Table 2c), and audit-log querying (Table 2d). We specifically measured the time of each access request and each audit-log query. To ensure the robustness and statistical significance of our findings, experiments were repeated 30 times. Additionally, we applied unpaired t-tests to all inter-institutional comparisons, with a significance threshold at 0.05.

Results

Smart contract deployment

The initial deployment of smart contracts was launched from Institution 1, which then shared the addresses of these smart contracts with all participating sites, representing a one-time setup cost for each experiment. The entire duration to deploy 1 Connector, 3 Institution, 3 Researcher Management, 3 Data Concierge, 1 Log, and 1 Date Time contracts, along with the branching of their interconnections, took on average 36.035 s with a standard deviation of 2.032 s over 30 experiments. On a per-contract basis, the average smart contract deployment time was 2.117 s with a standard deviation of 0.123 s.

Metadata initialization

Figure 6A presents the overall metadata import times for the three institutions. The system took a maximum import time of 146.620 s at Institution 3 for both dataset and researcher's certificate metadata. On the other hand, Institution 1 took the shortest time of 135.265 s. Comprehensive details of the data import times for each of the three sites are demonstrated in Table 4a.

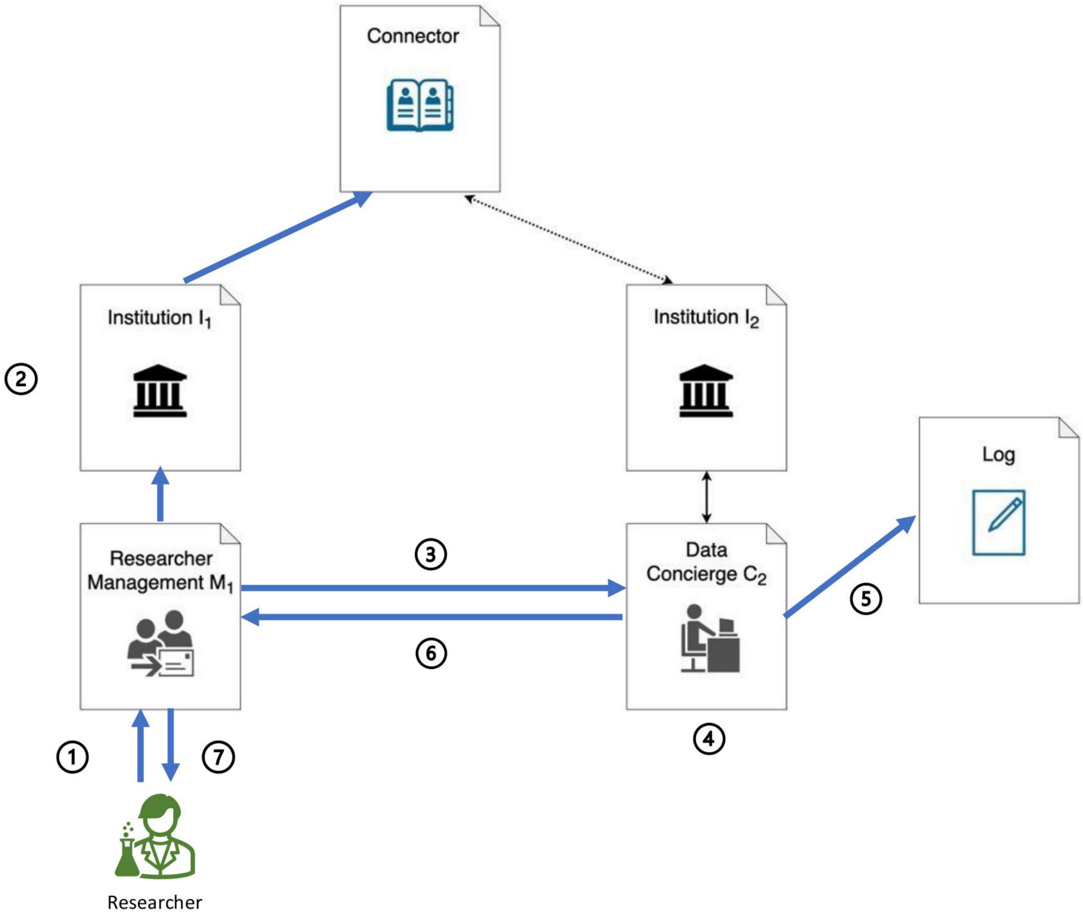

Figure 4. Workflow of researcher requests and smart contract interactions between two institutions. A researcher from Institution 1 requests a dataset from Institution 2's data concierge office. *Step 1*. Researcher submits a request via Institution 1's Researcher Management 1 smart contract. *Step 2*. This contract retrieves addresses of Connector and Institution 2's Data Concierge 2 smart contract. *Step 3*. Data request is sent to Data Concierge 2 smart contract. *Step 4*. Data Concierge 2 smart contract verifies training certificate and DUA information. *Step 5*. Data Concierge 2 smart contract records the request history in the Log smart contract. *Step 6*. Decision is relayed back to Researcher Management 1 smart contract. *Step 7*. Researcher receives a response.

Table 3. Statistics of the synthetic data.

Institution	Metadata		Researcher access request	Audit-log query
	Number of researchers	Number of datasets	Number of researcher access requests	Number of log queries
1	20	10	350	363 072
2	20	10	266	
3	20	10	411	
Total	60	30	1027	

Researcher request processing

The total runtime for all researcher requests varied across institutions, with Institution 3 experiencing the longest duration of 984.385 s and Institution 2 the shortest at 574.783 s. Per institution, the average time for a researcher request ranged approximately from 2.161 to 2.395 s. Detailed results of the researcher requests for all three sites are presented in Table 4b. These variations are further depicted in Figure 6B, which illustrates the time taken per researcher request at each of the three institutions.

Audit-Log querying

A total of 363 072 audit-log queries were performed at all three institutions. The overall runtime for these queries spanned a range from 399.223 to 417.933 s per site. This range is further explored in Figure 6C, which compares the per query time across the 3 sites. Given our exhaustive testing approach of all combinations, many of these queries did not yield any output. For queries with non-empty output (queries that yielded at least one record), the average return time varied from 2.613 to 2.708 ms. For queries with empty output (queries that did not yield any returned records), the average return time varied from 0.962 to 1.009 ms. Detailed results of the audit-log queries for each site are detailed in Table 4c.

Discussion

Findings in each workflow

In general, variations were observed in time measurements of metadata initialization, researcher request processing, and

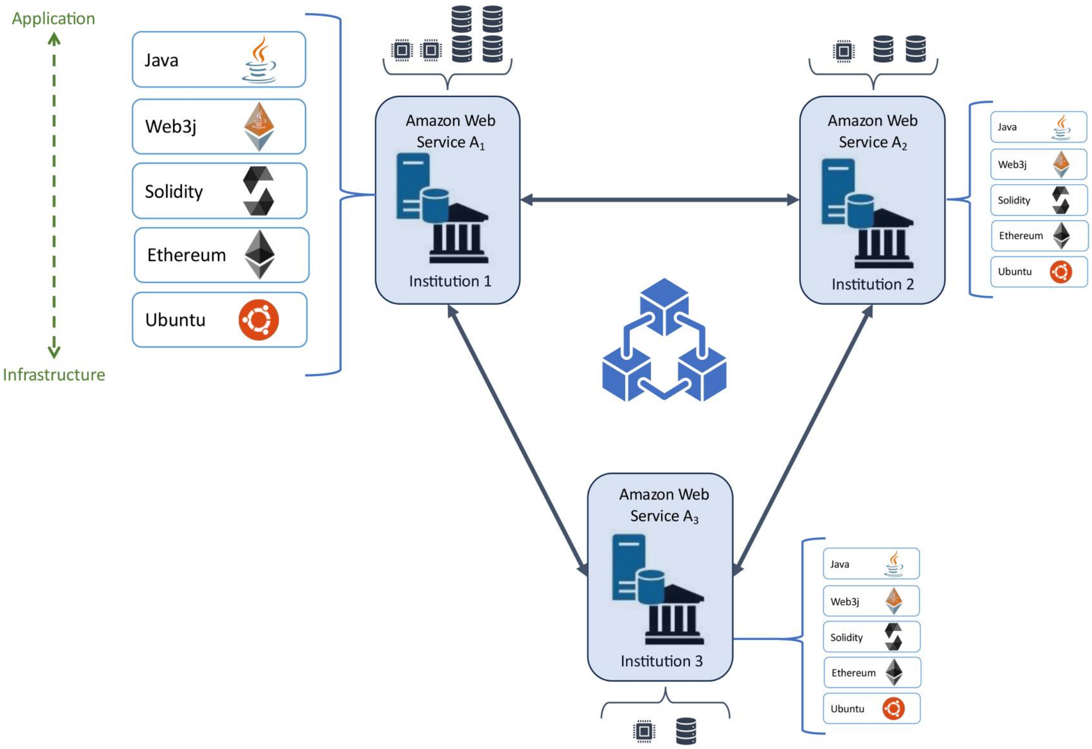

Figure 5. Implementation architecture of the system. The system was hosted on a 3-node blockchain network with three Amazon Web Service (AWS) virtual machines of different computational resources. At every node, the same technology stack from infrastructure to application level was implemented, comprising the underlying Ubuntu operating system of the virtual machines, the Ethereum blockchain, and the programming language Solidity to enable the smart contract components. The Web3j library was used to relay communication between on-chain and off-chain modules, and a Java-based backend method was developed to facilitate data submission and retrieval.

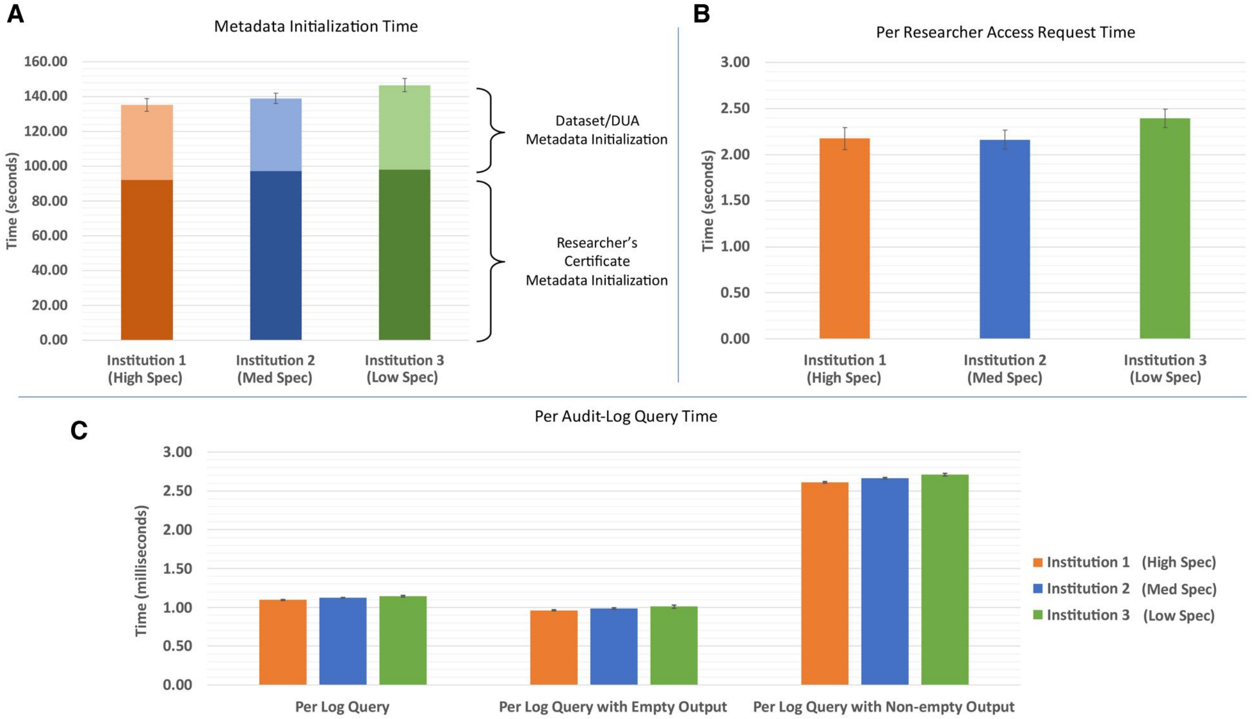

Figure 6. Evaluation of system performance across three emulated institutions. The standard deviation for each measured value is represented as an error bar. (A) Overall metadata initialization time, including import times for researchers' certificate metadata and dataset metadata. (B) Per researcher access request time. (C) Per audit-log query time.

Table 4. Detailed running time results of (a) metadata initialization, (b) researcher access requests, and (c) audit-log queries, at three institutions.

Statistics		Institution 1	Institution 2	Institution 3
(a) Metadata initialization	Overall	135.265 (6.393)	139.044 (5.400)	146.620 (5.897)
	Overall researcher's certificate metadata import	92.250 (4.806)	97.161 (4.378)	98.095 (4.719)
	No. of certificates*	29	34	27
	Per certificate	3.181 (0.166)	2.858 (0.129)	3.633 (0.175)
	Overall dataset metadata import	43.015 (2.342)	41.883 (1.635)	48.525 (2.933)
	No. of datasets*	10	10	10
	Per dataset	4.301 (0.234)	4.188 (0.163)	4.853 (0.293)
(b) Researcher access request processing	Overall	760.895 (41.522)	574.783 (27.401)	984.385 (41.057)
	No. of researchers' requests*	350	266	411
	Per request	2.174 (0.119)	2.161 (0.103)	2.395 (0.100)
	Per successful request	2.183 (0.141)	2.148 (0.103)	2.399 (0.122)
	Per denied request	2.172 (0.116)	2.164 (0.107)	2.394 (0.100)
(c) Audit-log querying	Overall	399.223 (1.842)	409.094 (3.307)	417.933 (3.558)
	No. of log queries*	363,072	363,072	363,072
	Per query (milliseconds)	1.096 (0.005)	1.123 (0.009)	1.147 (0.010)
	Per query, with empty output (milliseconds)	0.962 (0.005)	0.987 (0.008)	1.009 (0.009)
	Per query, with non-empty output (milliseconds)	2.613 (0.010)	2.661 (0.017)	2.708 (0.018)

All times (except per query results) are measured in seconds, with standard deviations provided in parentheses.

* Counts.

audit-log record querying across the three institutions, which may be attributed to the difference in specifications of their respective hardware. The overall system performance is determined by Institution 3 (the VM with the lowest specifications).

For metadata initialization, Institution 2 has the fastest per-metadata initialization time for both researcher's certificate metadata and dataset metadata. Specifically, with regards to per certificate import speed, Institution 2 (2.858 s) demonstrated a 10.16% faster performance compared to Institution 1 (*P*-value of 7.75×10^{-11}) and a 21.34% faster performance than Institution 3 (*P*-value of 1.11×10^{-18}). Similarly, in the case of per dataset metadata import, Institution 2 (4.188 s) was marginally faster than Institution 1 at 2.63% (*P*-value of .043) and 13.69% faster than Institution 3 (*P*-value of 3.27×10^{-13}).

For researcher access request processing, given the variation in the total number of requests per institution, we utilized the time metric of per researcher request for comparative analysis. On this basis, Institution 1 processed requests 9.23% faster than Institution 3 (*P*-value of 3.90×10^{-7}). However, the difference in request times between Institution 1 and 2 was minimal (*P*-value of .117), which might be due to the fact that the ratios of successful and denied requests might differ at each site. The *t*-test analysis of per successful request (*P*-value of .049) and per denied request (*P*-value of .407) between Institution 1 and 2 confirmed that there was no significant time difference in the per researcher request.

For audit-log querying, the variation observed in overall log query time follows the differences in computational specification of the three institutions: Institution 1 had the fastest performance, followed by Institution 2 and 3. It was also observed that the majority of the time consumed in this auditing function might be spent on constructing the string value of the output, which may explain why queries without empty output (ie, denied) were faster than those yielding non-empty output (ie, approved). The results showed that Institution 1 was approximately 2.41% faster than Institution 2 (*P*-value of 1.43×10^{-17}) and 4.48% faster than Institution 3 (*P*-value of 2.24×10^{-22}). Notably, queries resulting in empty output were observed to be significantly faster than those with non-empty output (*P*-value of 3.63×10^{-170}), which also has greater standard deviation. Despite the differences between queries with empty and non-empty output, we were able to achieve sufficiently fast query times at the millisecond range.

Findings in comparative analysis

The comparative analysis among the three institutions reveals that the hardware capacity and computational power can influence the time a VM takes to complete a given task. This was particularly evident in audit-log querying which was executed locally on each VM. The results mirror the computational capacity hierarchy of high (Institution 1), medium (Institution 2), and low (Institution 3).

Nevertheless, it is possible that once a VM's specifications reach a certain threshold ("medium" or higher in computational power), further improvements in performance become marginal, at which point other factors such as network speed begin to exert a more pronounced effect on results. In our experiments, both metadata initialization and researcher access request processing involve writing to the blockchain and possibly interacting with other smart contracts. Consequently, network speed may become a crucial factor in these processes, potentially accounting for the better performance of Institution 2 in these specific areas compared to Institution 1.

Our choice of a private blockchain implementation utilizing a PoA protocol reduces the impact of computational capacity, which is less critical than on a public blockchain where extensive computation is essential for maintaining speed. Thus, even with limited computational power as seen in Institution 3, the system managed to maintain a relatively quick average response time of 2.395 s per researcher request. This efficiency marks an advancement from the traditional manual email method, demonstrating the system's ability to reduce response times from potentially hours or days to mere seconds, even under varied computational capacities.

Limitations

There are several limitations with our current system design and experiments as follows.

1) To demonstrate the feasibility of a blockchain-based LDS management system, we evaluated our method on a fixed-size network. A recent study demonstrated that system performance remained stable as the number of network nodes increased from 3 to 5,[43] suggesting that adding more institutions could enhance the aggregated computational power. However, the full impact of node scalability on system speed, stability, and overall performance remains to be explored, particularly with a substantial increase in nodes.
2) Beyond the number of institutional sites, the volume of data requests could also impact the scalability of our system. While our system has been evaluated with over 1000 data requests, larger or less-uniformly-distributed requests across institutions could affect performance. For example, one institution may have many users submitting data requests simultaneously. To address such scenarios, potential improvements could include concurrent or grouped submission of requests or deploying multiple nodes per institution to better manage larger request volumes.
3) Another limitation of our evaluation is the use of synthetic data. Real data requests may introduce additional metadata elements that our current prototype has not accounted for. We have yet to collaborate with various institutions to gather detailed metadata fields, and to develop a generalized metadata coding system that can accommodate a broader range of real-world datasets.
4) Our system was tested on three AWS virtual machines to simulate three institutional sites. However, real-world institutions might use different cloud providers, such as Google Cloud Platform or Microsoft Azure, each of which may have unique technical configurations that could impact network latency and bandwidth. Additionally, the AWS servers in our study were likely confined to a single region, which may not accurately reflect the diversity of institutions operating across multiple regions or globally. Previous research has shown that a cross-cloud infrastructure can achieve relatively consistent performance across platforms.[24] Future improvement of our system could involve evaluating both cross-cloud architectures and geographically distributed setups to better reflect real-world scenarios.
5) Our system currently utilizes Ethereum as its blockchain infrastructure. We have yet to explore other blockchain frameworks (eg, Hyperledger Fabric or Corda) for their varying scalability, privacy features, and security mechanisms that could better support the protection of sensitive biomedical data.
6) While the core components of our method, specifically the smart contract codes, are available on GitHub to facilitate easier deployment and replication, we have yet to develop a fully hardened software and build a comprehensive codebase or Docker-based implementation of the system.
7) Our system currently prioritizes secure and efficient data access within the existing framework, which minimizes the attack surface by avoiding the direct sharing of full data files. As such, we have yet provided direct access to data content beyond data links. Integrating the InterPlanetary File System (IPFS) for distributed storage and encrypted data transmission is a potential enhancement.[36,50] Data stored in IPFS may be split into smaller encrypted chunks before being distributed across the network. Files are then retrieved based on their cryptographic hash; only authorized users with the appropriate decryption key are able to reassemble and decrypt the data. This ensures that even if the hash or encrypted chunks are exposed, the underlying sensitive data remains secure and inaccessible without the key.[51]
8) In terms of enhancing integrity and security, we have yet to restrict system access to institutional personnel through institutionally-verified sign-on procedures. Since every action on the blockchain is recorded immutably, implementing a credential-based access control mechanism can connect these activities to specific users, which could improve traceability and ensure the integrity/security of data transactions within the system.
9) Direct comparison with current human-based processes could provide valuable insights into the quantitative improvement and efficiency gained through our blockchain-based method. We have yet to collaborate with data concierge staff to empirically measure the end-to-end time for dataset sharing across institutions, using both manual protocols and our system. We also aim to develop a web-based user interface and conduct quantitative studies to assess its usability among researchers, institutions, and concierge staff.

Conclusions

Our system demonstrates a viable prototype leveraging blockchain networks and smart contracts to streamline the LDS sharing process with approximately 2 s of processing and immutably storing each request on chain. It addresses challenges such as the elimination of SPoF and unauthorized data modifications, while ensuring the intuitive auditability and transparency of the data access trail. The system has been evaluated across multiple sites in a pragmatic computing environment, featuring varying computational capacities across sites. The evaluation demonstrates the system's efficiency; as even the computational setup with the lowest specifications was reasonably sufficient, the system has shown its definite edge over traditional manual email-based methods.

Our blockchain-based automated data sharing system may also revolutionize the way clinical research data is shared among researchers, thereby facilitating stronger cross-institutional collaboration and expediting advancements in biomedical studies. Specifically, clinical researchers are provided with quicker and more reliable access to diverse datasets, leading to potential improvement in diagnostic accuracy and more personalized patient treatment plans. Meanwhile, patients may benefit from the immutable access history of their health information, as supported by transparent audit logs to assure that their data are only accessed by granted parties. Moreover, auditors can review the access audit trail to ensure the whole data request process is conforming to policies and regulations. This innovative approach can both accelerate the speed of biomedical research and clinical studies and establish a stronger foundation for how medical data is managed and utilized.

Acknowledgements

The authors would like to thank Michael Hogarth, MD, for supporting cloud computing environment.

Author contributions

YY contributed to methodology, software, validation, formal analysis, investigation, writing (original draft, review & editing), and visualization. ME contributed to methodology, software, investigation, and writing (review & editing). AP contributed to the conceptualization, data curation, methodology, and writing (review & editing). JP, BJ, and KP contributed to methodology and software. T-TK contributed to the conceptualization, methodology, software, validation, formal analysis, investigation, resources, visualization, supervision, project administration, funding acquisition, and writing (review & editing).

Funding

The authors were funded by the U.S. National Institutes of Health (NIH) (R01EB031030, R01HG011066, and T15LM011271) and the Graduate Division San Diego Matching Fellowship associated with San Diego Biomedical Informatics Education & Research (SABER) NIH National Library of Medicine (NLM) grant T15LM011271. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. This research benefited from the use of credits from the National Institutes of Health (NIH) Cloud Credits Model Pilot, a component of the NIH Big Data to Knowledge (BD2K) program.

Conflicts of interest

None declared.

Data availability statement

The data underlying this article and smart contracts are available at: https://github.com/graceyufei/LDS-Request-Management.

References

1. Margolis R, Derr L, Dunn M, et al. The National Institutes of Health's Big Data to Knowledge (BD2K) initiative: capitalizing on biomedical big data. *J Am Med Inform Assoc*. 2014;21:957-958. https://doi.org/10.1136/amiajnl-2014-002974
2. Tedersoo L, Küngas R, Oras E, et al. Data sharing practices and data availability upon request differ across scientific disciplines. *Sci Data*. 2021;8:192. https://doi.org/10.1038/s41597-021-00981-0
3. Office for Civil Rights. A decision tool: Limited Data Set (LDS). HHS.gov. 2022. Accessed April 11, 2024. https://www.hhs.gov/hipaa/for-professionals/special-topics/emergency-preparedness/limited-data-set/index.html
4. Digital Communications Division. What is phi? HHS.gov. 2021. Accessed April 11, 2024. https://www.hhs.gov/answers/hipaa/what-is-phi/index.html
5. U.S. Department of Labor. *Guidance on the Protection of Personal Identifiable Information*. Accessed April 11, 2024. https://www.dol.gov/general/ppii
6. Clarity. *Limited Data Set: The Complete Guide*. Accessed May 7, 2024. https://www.clarity-ventures.com/glossary/limited-data-set#:~:text=Why%20Is%20a%20Limited%20Data,treatments%2C%20and%20advance%20medical%20research
7. National Institutes of Health. *HIPAA Privacy Rule and its Impacts on Research*. Accessed December 7, 2023. https://privacyruleandresearch.nih.gov/pr_08.asp
8. Office for Civil Rights. *Research*. HHS.gov. 2021. Accessed April 11, 2024. https://www.hhs.gov/hipaa/for-professionals/special-topics/research/index.html
9. CITI Program. *Research, Ethics, and Compliance Training*. Accessed December 7, 2023. https://about.citiprogram.org/
10. Office for Civil Rights. *A Decision Tool: Data Use Agreement*. HHS.gov. 2022. Accessed November 9, 2024. https://www.hhs.gov/hipaa/for-professionals/special-topics/emergency-preparedness/data-use-agreement/index.html
11. Office of Research Integrity. Data Ownership. Accessed April 11, 2024. https://ori.hhs.gov/content/Chapter-6-Data-Management-Practices-Data-ownership
12. Champieux R, Solomonides A, Conte M, et al. Ten simple rules for organizations to support research data sharing. *PLoS Comput Biol*. 2023;19:e1011136. https://doi.org/10.1371/journal.pcbi.1011136
13. Tyner C. *Manual Process vs Automated Process: A Comparison Guide*. Comidor. 2019. Accessed April 11, 2024. https://www.comidor.com/blog/business-process-management/a-comparison-of-manual-and-automated-workflow-processes/
14. Gumrukcu E, Arsalan A, Muriithi G, et al. Impact of cyber-attacks on EV charging coordination: The case of single point of failure. *2022 4th Global Power, Energy and Communication Conference (GPECOM)*. Published Online First: 14 June 2022. https://doi.org/10.1109/gpecom55404.2022.9815727
15. Kuo T-T, Kim H-E, Ohno-Machado L. Blockchain distributed Ledger Technologies for Biomedical and Health Care Applications. *J Am Med Inform Assoc*. 2017;24:1211-1220. https://doi.org/10.1093/jamia/ocx068
16. Lawler R. Amazon's server outage broke fast food apps like McDonald's and Taco Bell. *The Verge*. 2023. Accessed April 11, 2024. https://www.theverge.com/2023/6/13/23759857/amazon-aws-down-outage-taco-bell-mcdonalds-burger-king
17. Thomas D. Pa. Firm alleges ex-partners 'secretly planned' to join Armstrong Teasdale. *Westlaw Today*. 2021. Accessed January 16, 2024. https://today.westlaw.com/Document/I1c80a1d070ac11eb-b555947e94fe83f6/View/FullText.html?transitionType=SearchItem&contextData=%28sc.Default%29&firstPage=true
18. U.S. Attorney's Office, Western District of Washington. Former Seattle Tech worker convicted of wire fraud and computer intrusions. Former Seattle tech worker convicted of wire fraud and computer intrusions | United States Department of Justice. 2022. Accessed January 16, 2024. https://www.justice.gov/usao-wdwa/pr/former-seattle-tech-worker-convicted-wire-fraud-and-computer-intrusions
19. Jahansoozi J. Organization-stakeholder relationships: exploring Trust and transparency. *Journal of Management Development*. 2006;25:942-955. https://doi.org/10.1108/02621710610708577
20. Lacson R, Yu Y, Kuo T-T, et al. Biomedical blockchain with practical implementations and quantitative evaluations: a systematic review. *J Am Med Inform Assoc*. 2024;31:1423-1435. https://doi.org/10.1093/jamia/ocae084
21. Dimitrov DV. Blockchain applications for healthcare data management. *Healthc Inform Res*. 2019;25:51-56. https://doi.org/10.4258/hir.2019.25.1.51
22. Yaqoob I, Salah K, Jayaraman R, et al. Blockchain for healthcare data management: Opportunities, challenges, and future recommendations. *Neural Comput Applic*. 2022;34:11475-11490. https://doi.org/10.1007/s00521-020-05519-w
23. Kuo T-T, Jiang X, Tang H, et al. The evolving privacy and security concerns for genomic data analysis and sharing as observed from the IDASH competition. *J Am Med Inform Assoc*. 2022;29:2182-2190. https://doi.org/10.1093/jamia/ocac165
24. Kuo T-T, Pham A, Edelson ME, R2D2 Consortium, et al. Blockchain-enabled immutable, distributed, and highly available clinical research activity logging system for Federated covid-19 data

analysis from multiple institutions. *J Am Med Inf Assoc*. 2023;30:1167-1178. https://doi.org/10.1093/jamia/ocad049
25. Medicalchain. Accessed January 16, 2024. https://medicalchain.com/en/
26. Kuo T-T, Bath T, Ma S, et al. Benchmarking blockchain-based gene-drug interaction data sharing methods: a case study from the iDASH 2019 secure genome analysis competition blockchain track. *Int J Med Inform*. 2021;154:104559. https://doi.org/10.1016/j.ijmedinf.2021.104559
27. Kuo T-T, Jiang X, Tang H, et al. iDASH Secure Genome Analysis Competition 2018: blockchain genomic data access logging, homomorphic encryption on GWAS, and DNA segment searching. *BMC Med Genom*. 2020;13:98. https://doi.org/10.1186/s12920-020-0715-0
28. BurstIQ. Accessed January 16, 2024. https://burstiq.com/
29. Li MM, Kuo T-T. Previewable contract-based on-chain X-ray image sharing framework for Clinical Research. *Int J Med Inform*. 2021;156:104599. https://doi.org/10.1016/j.ijmedinf.2021.104599
30. ProCredEx. Accessed January 16, 2024. https://procredex.com/
31. Christidis K, Devetsikiotis M. Blockchains and smart contracts for the internet of things. *IEEE Access*. 2016;4:2292-2303. https://doi.org/10.1109/access.2016.2566339
32. Curbera F, Dias DM, Simonyan V, et al. Blockchain: an enabler for healthcare and life sciences transformation. *IBM J Res & Dev*. 2019;63:8:1-8:9. https://doi.org/10.1147/jrd.2019.2913622
33. Pincheira M, Donini E, Vecchio M, et al. A decentralized architecture for trusted dataset sharing using smart contracts and distributed storage. *Sensors (Basel)*. 2022;22:9118. https://doi.org/10.3390/s22239118
34. Purohit S, Calyam P, Alarcon ML, et al. Honestchain: consortium blockchain for protected data sharing in health information systems. *Peer Peer Netw Appl*. 2021;14:3012-3028. https://doi.org/10.1007/s12083-021-01153-y
35. Tellew J, Kuo T-T. Certificatechain: decentralized healthcare training certificate management system using blockchain and smart contracts. *JAMIA Open*. 2022;5:ooac019. https://doi.org/10.1093/jamiaopen/ooac019
36. Nyaletey E, Parizi RM, Zhang Q, et al. BlockIPFS—blockchain-enabled interplanetary file system for forensic and Trusted Data Traceability. *2019 IEEE International Conference on Blockchain (Blockchain)*. Published Online First: July 2019. https://doi.org/10.1109/blockchain.2019.00012
37. Yu H, Sun H, Wu D, Kuo T-T. Comparison of smart contract blockchains for healthcare applications. In: *AMIA Annual Symposium: American Medical Informatics Association*, Bethesda, MD, 2019.
38. Kuo T-T, Zavaleta Rojas H, Ohno-Machado L. Comparison of blockchain platforms: a systematic review and healthcare examples. *J Am Med Inform Assoc*. 2019;26:462-478. https://doi.org/10.1093/jamia/ocy185
39. Buterin V. A next-generation smart contract and decentralized application platform. *White Paper*. 2014;3:2-1.
40. IBM. What is Hyperledger Fabric? Accessed September 30, 2024. https://www.ibm.com/topics/hyperledger
41. Corda. Accessed September 30, 2024. https://corda.net
42. Pham A, Minihan JRC, Augustine A, Kuo TT. Corda blockchain for interoperable insurance billing. *AMIA Informatics Summ*it, 2023.
43. Pham A, Edelson M, Nouri A, Kuo T-T. Distributed management of patient data-sharing informed consents for clinical research. *Comput Biol Med*. 2024;180:108956. https://doi.org/10.1016/j.compbiomed.2024.108956
44. Poa private chains. GitHub. Accessed January 17, 2024. https://github.com/ethereum/guide/blob/master/poa.md
45. Guegan D. *Public Blockchain versus Private blockchain*. 2017. Accessed April 11, 2024. https://shs.hal.science/halshs-01524440 (accessed 11 Apr2024).
46. Solidity. Solidity 0.8.13 documentation. Accessed December 7, 2023. https://docs.soliditylang.org/en/latest/
47. Go Ethereum. Ethereum. Accessed December 7, 2023. https://geth.ethereum.org/
48. Remix. Remix—Ethereum IDE. Accessed December 7, 2023. https://remix.ethereum.org/
49. Web3j: Lightweight java and Android Library for integration with Ethereum clients. GitHub. Accessed December 7, 2023. https://github.com/web3j/web3j
50. Naz M, Al-Zahrani FA, Khalid R, et al. A secure data sharing platform using blockchain and interplanetary file system. *Sustainability*. 2019;11:7054. https://doi.org/10.3390/su11247054
51. Concepts. IPFS Docs. Accessed October 9, 2024. https://docs.ipfs.tech/concepts/

Journal of the American Medical Informatics Association, 2025, 32, 296–307
https://doi.org/10.1093/jamia/ocae288
Research and Applications