Journal of the American Medical Informatics Association, 30(6), 2023, 1167–1178
https://doi.org/10.1093/jamia/ocad049
Advance Access Publication Date: 14 March 2023
Research and Applications

Research and Applications

Blockchain-enabled immutable, distributed, and highly available clinical research activity logging system for federated COVID-19 data analysis from multiple institutions

Tsung-Ting Kuo[1], Anh Pham[1], Maxim E. Edelson[2], Jihoon Kim[1], Jason Chan[3], Yash Gupta[4], Lucila Ohno-Machado[1,5,6] and The R2D2 Consortium

[1]UCSD Health Department of Biomedical Informatics, University of California San Diego, La Jolla, California, USA, [2]Department of Computer Science and Engineering, University of California San Diego, La Jolla, California, USA, [3]Poway High School, Poway, California, USA, [4]Canyon Crest Academy, San Diego, California, USA, [5]Division of Health Services Research & Development, VA San Diego Healthcare System, San Diego, California, USA and [6]Biomedical Informatics and Data Science, Yale School of Medicine, New Haven, Connecticut, USA

Tsung-Ting Kuo, Anh Pham, and Maxim E. Edelson contributed equally to this work.

*The full list of the authors is included in the Supplementary Appendix.

Corresponding Author: Dr Tsung-Ting Kuo, PhD, UCSD Health Department of Biomedical Informatics, University of California San Diego, 9500 Gilman Dr, La Jolla, San Diego, CA 92093, USA; tskuo@ucsd.edu

Received 26 August 2022; Revised 7 March 2023; Editorial Decision 8 March 2023; Accepted 11 March 2023

ABSTRACT

Objective: We aimed to develop a distributed, immutable, and highly available cross-cloud blockchain system to facilitate federated data analysis activities among multiple institutions.
Materials and Methods: We preprocessed 9166 COVID-19 Structured Query Language (SQL) code, summary statistics, and user activity logs, from the GitHub repository of the Reliable Response Data Discovery for COVID-19 (R2D2) Consortium. The repository collected local summary statistics from participating institutions and aggregated the global result to a COVID-19-related clinical query, previously posted by clinicians on a website. We developed both on-chain and off-chain components to store/query these activity logs and their associated queries/results on a blockchain for immutability, transparency, and high availability of research communication. We measured run-time efficiency of contract deployment, network transactions, and confirmed the accuracy of recorded logs compared to a centralized baseline solution.
Results: The smart contract deployment took 4.5 s on an average. The time to record an activity log on blockchain was slightly over 2 s, versus 5–9 s for baseline. For querying, each query took on an average less than 0.4 s on blockchain, versus around 2.1 s for baseline.
Discussion: The low deployment, recording, and querying times confirm the feasibility of our cross-cloud, blockchain-based federated data analysis system. We have yet to evaluate the system on a larger network with multiple nodes per cloud, to consider how to accommodate a surge in activities, and to investigate methods to lower querying time as the blockchain grows.
Conclusion: Blockchain technology can be used to support federated data analysis among multiple institutions.

Key words: COVID-19, electronic health record, blockchain distributed ledger technology, clinical information systems, decision support systems

BACKGROUND AND SIGNIFICANCE

Federated COVID-19 data analysis

The global COVID-19 pandemic has changed the field of healthcare in profound ways,[1] including how healthcare professionals gather and communicate information given the exponential increase of available data and research works related to the topic.[2] For example, natural language processing (NLP) and deep neural networks have helped automate information retrieval tasks, such as scanning large bodies of biomedical literature with specific human-language questions in mind, and identifying the corresponding text-based answers from this pool of established knowledge.[3,4] Another task that also relies on aggregated information is federated data analytics, which facilitates ongoing research by combining a larger sample size from multiple sources, thereby potentially increasing statistical power and generalizability of findings.[5,6] Specifically in the context of COVID-19, a rapidly spreading infectious disease with vast socioeconomic and geographic impact, the practice of gathering information across institutions and regions in a timely manner is desired,[7,8] thus leading to the formation of data consortia for a federated data analysis. These consortia facilitate achieving a larger sample size from multiple sources, thereby potentially increasing statistical power and generalizability of findings without sharing individual level data in each participating institution.[9,10] Meanwhile, rapid communication among participating members of data consortia is critical to enhance clinical and public health efforts, as they help analyze broader observational data when compared to registries, on which large-scale analyses can be carried out to interrogate clinical progression, therapeutic efficacy, and outcomes.[11–13]

The consortium for COVID-19 federated data analysis

The Reliable Response Data Discovery for COVID-19 (R2D2) Consortium[14] was established in 2020 and funded by the Gordon and Betty Moore Foundation, consisting of 13 US health institutions—Cedars Sinai (CS), Emory University (EMORY), Georgetown University (GT), San Mateo Medical Center (SMMC), University of California Davis (UCD), University of California Irvine (UCI), University of California Los Angeles (UCLA), University of California San Diego (UCSD), University of California San Francisco (UCSF), University of Colorado Anschutz Medical Campus (CUA), University of South California (USC), the University of Texas (UT) Health Science Center at Houston, Veterans Affairs (VA)—and Ludwig Maximilian University (LMU) in Germany. The R2D2 Consortium allowed prompt exchange of summary statistics, such as percentage among participating institutions, and took advantage of a substantially large patient pool to empower research while still maintaining patient privacy since patient-level data never left their original sites.[14] The Consortium workflow started from a researcher who had a clinical question about COVID-19 (eg, What percentage of patients needed dialysis?), who then submitted the text into a web portal (https://covid19questions.org), and ended with the researcher receiving an email notification about a response (eg, 15% receive dialysis at a particular institution).

Specifically, R2D2 consortium data were used to facilitate ongoing research and patient care analytics in a privacy-preserving manner.[14] Some of the responded questions include: (1) Among adult COVID-19 patients who were hospitalized excluding the Intensive Care Unit and discharged alive, how many returned to the hospital within a week, either to the Emergency Room or for another hospital stay? The answer was 8.64%, or 279 patients out of 3230 ones from 10 health systems. (2) Among adults hospitalized with COVID-19, how does the in-hospital mortality rate compare per subgroup, such as age, ethnicity, sex, and race? The result showed that the mortality rate was higher in older age groups (26% in age group 81 years or older vs 1% below age 40 years), Non-Hispanic ethnicity (12.32%), male sex (13.32%), and white race (12.25%).

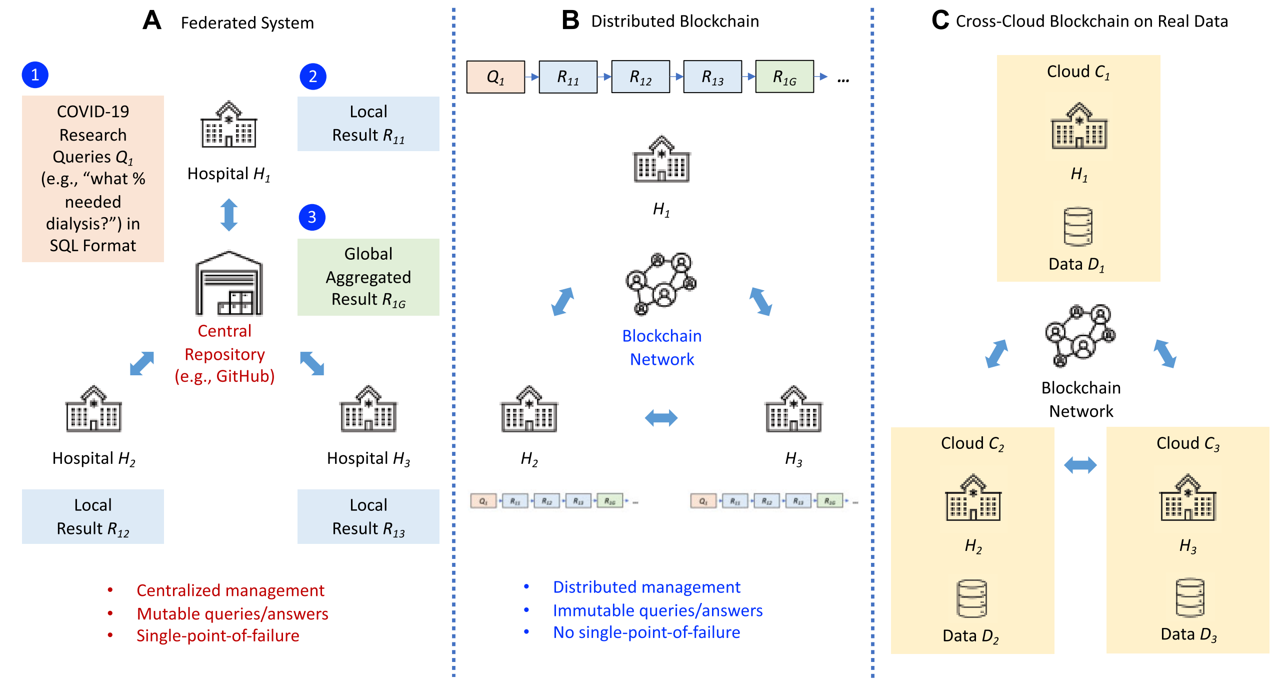
In the backend, the whole process entails several steps. First, the received question would be assigned to one of the Consortium members' sites (Lead Site) and reviewed by clinicians who would work with the database analyst to translate the clinical question from text format to Structured Query Language (SQL) code, run the code, verify the resulting local summary statistics, and release the SQL code to the Consortium Hub. At the other site (Responding Site), the database analyst would download the SQL code written by Lead Site, review, and make custom changes to the code including Export-Translate-Load (ETL) process of local Electronic Health Record (EHR) systems, local data harmonization, and local data quality assurance in an iterative process of communication with the Lead Site and local clinicians.

The full details of the workflow are described by Kim et al.[14] An example of how R2D2 and the consortium model might enable original research is shown in Figure 1A. Upon receiving a COVID-19 research query in SQL format,[15] each participating hospital would use this SQL code to search their local database for the corresponding numerical result, then submit their local statistics to a central repository that aggregated individual numbers to yield a global statistics result.

Challenges of using GitHub to share SQL code and summary statistics

R2D2 Consortium used a website-based GitHub,[14] a widely popular platform for a distributed version control system in both academia and industry, to share SQL codes and query results of summary statistics.[16] As the role of Lead Site was a resource-intensive task, each institution took a turn based on clinical expertise and subject matter proficiency. This required any consortium member to be able to submit initial SQL code, modify it, and merge with other codes at any given time. Thus, restricting privileges to specific members was not in the consortium's best interest, as workflow bottlenecks could easily happen should one have to defer their work and wait for someone else with higher access privilege. Even for well-adapted services such as GitHub, it is not trivial to customize access control permissions and/or establish proper topology without defaulting to a hierarchical setting in which some members bear more control than others.[17] This led the consortium to decide early on to democratize repository read/write permissions to all members, given the urgency of obtaining clinical knowledge about rapidly spreading COVID-19, the unprecedented nature of the pandemic, and the ultimate goal of the consortium to facilitate rapid research progress across institutions. However, the use of GitHub to share SQL code and summary statistics faced several limitations.

First, it is not possible to ensure the *preservation* of the recorded SQL code and summary statistics results. When all members have read/write permission to the GitHub repository, where SQL codes and their corresponding local summary statistics are gathered, there is a chance some institution may accidentally delete files without being timely noticed, causing the loss of intermediate and/or final results of the federated data analyses.



A Federated System
1
COVID-19 Research Queries Q_1 (e.g., "what % needed dialysis?") in SQL Format
Hospital H_1
2
Local Result R_{11}
3
Global Aggregated Result R_{1G}
Central Repository (e.g., GitHub)
Hospital H_2
Hospital H_3
Local Result R_{12}
Local Result R_{13}
• Centralized management
• Mutable queries/answers
• Single-point-of-failure
B Distributed Blockchain
Q_1 R_{11} R_{12} R_{13} R_{1G} ...
H_1
Blockchain Network
H_2
H_3
• Distributed management
• Immutable queries/answers
• No single-point-of-failure
C Cross-Cloud Blockchain on Real Data
Cloud C_1
H_1
Data D_1
Blockchain Network
Cloud C_2
H_2
Data D_2
Cloud C_3
H_3
Data D_3

Figure 1. Comparison of different COVID-19 data analysis systems. (A) Federated system.[14] This example system consists of 3 hospitals (H_1, H_2, and H_3) and a central repository (eg, GitHub). In Step (1), a researcher in a hospital (H_1) proposed a COVID-19 research query (Q_1). Next in Step (2), each hospital views and submits results (R_{11}, R_{12}, and R_{13}) based on local Electronic Health Record (EHR) data. Finally in Step (3), the results are aggregated to formulate the global results (R_{1G}) to the original query. No patient-level data are disseminated; thus, patient privacy is protected. However, all queries, local results, and global results are stored in the central repository, which presents a centralized management scheme, a single "root" administrative user (who may alter the research queries and results without being noticed), and a single-point-of-failure. (B) Distributed Blockchain. Using the peer-to-peer blockchain technology, no central repository is required, and everyone can see all "transactions (TXs)" (ie, queries, their local results, and global results) on the blockchain. This way, the system is fully distributed (no centralized management), immutable (no single user can change the transactions), and highly available (no single-point-of-failure), while it still protects patient privacy by only sharing aggregated data. (C) Cross-Cloud Blockchain on Real Data. Since each hospital may host their real-world COVID-19 data (D_1, D_2, and D_3) in different types of cloud computing environment, the design of the system should also take this into account.

Second, even when the preservation issue of SQL code and query results of summary statistics might be mitigated by the utilization of automatic backup and the nature of version control software, the "*root*" *administrative user* of the central repository still has the privilege to alter the submitted queries/results. While the standard practice is to avoid using "root" access, existing real-life events have proven that exploitative "root" overreach can still occur due to either noncompliant employees[18,19] or malicious attacks on inherent system vulnerability,[20,21] posing a genuine threat to network security.

Finally, the central repository (ie, GitHub) is a potential *single-point-of-failure*, as the whole system will stop working if this central repository becomes unavailable, as evidenced by previous incidents of severe disruptions and damages when a central system turns nonfunctional,[22–25] including redundancy failures in major databases.[26–28] In the specific instance of GitHub, server issues causing service disruptions have recently occurred.[29,30]

Blockchain technology to improve immutability, distributedness, and availability

To overcome some of the challenges outlined above, we explored blockchain, a practically immutable distributed technology with desirable technological features that may ease the time-sensitive pressure on data consortiums, by providing immutable SQL codes and query results of summary statistics, decentralized management without needing a central server, and highly available ledgering.

Blockchain technology,[31] which originated from the financial technology field and is now being proposed for various healthcare applications,[32,33] can be a solution to the centralized hub approach used to integrate data from various sites.[32,33] As shown in Figure 1B, in our prototype each COVID-19 research SQL code, local query results of summary statistics, and global (ie, aggregated) summary statistics result were stored on a blockchain, which is *immutable* because once data is recorded on chain it is very difficult, if not impossible, to change the ledgered data.[33] Beyond SQL code and summary statistics, the immutable timestamping mechanism of blockchain also ensures the chronological originality of SQL code posed, allowing for both in-network collaborations and protection of research idea ownerships. This feature may have value when we take into consideration that it is technically possible to amend GitHub history,[34] or timestamps of other file-transferring systems such as ext4 without leaving traces that can easily be noticed in time.[35]

Next, as no centralized server controls all resources, no single site can change the stored SQL code and summary statistics. If any institution leaves the consortium, the whole system could still function, as opposed to the scenario in which the institution that is in charge of integrating data leaves and the whole consortium halts operations until a new "main governing" role is established. This *decentralized* management capability is essential to support the consortium.

Also, based on the blockchain mechanism, each site had a full copy of the blockchain ledger. This in effect makes *high availability* a built-in part of blockchain without the need for intentional human designs, as compared to conventional database systems where redundancy requires conscious efforts yet may still fail.[26–28]

In short, the blockchain system is distributed, immutable, and remains highly available, and thus is suitable for record clinical research activities in the consortium. Meanwhile, the protection of patient privacy is still maintained in this design, given that only aggregated data, and not individual-level information, are shared. This makes blockchain as strong a candidate for privacy protection as any other federated systems. Additionally, proposed or proven cases of blockchain technology for clinical and genomic applications such as EHR access control,[36,37] privacy-preserving modeling,[38–43] genomic access logging,[44] gene-drug interaction data sharing,[45] clinical image sharing,[46] training certificates,[47] and patient data sharing consents[48] suggest that the use of blockchain networks may support consortium communications, adding its inherent advantages of immutability, distributedness, and high availability to current solutions.

Objective

We aimed to develop an immutable, distributed, and highly available blockchain system prototype on which cross-cloud, real-world COVID-19 clinical research communications among data-contributing members of a health consortium were logged and made searchable.

Contributions

We showcased the technical feasibility of a blockchain-based system, which could store and query the activities of the federated clinical research data analysis in an immutable, distributed, and highly available way. In particular, our work provided a robust architecture in which federated analysis using consortium data can be carried out efficiently and securely, allowing for secured collaboration on a larger aggregated sample size without subjecting patient privacy to external risks, and ultimately leading to meaningful research and/or quality care improvement in public health.

MATERIALS AND METHODS

To demonstrate the feasibility of a blockchain-based system for health data consortiums, we constructed a blockchain-based platform in which users could disseminate their research queries (in SQL format) to participating sites, provide corresponding results, and track activities through a search function that allows location of specific files meeting the search criteria (Figure 1C).

Data

During the internal workflow of the R2D2 Consortium, we stored the SQL codes in .sql format (instead of natural language texts) and site-level summary statistics in .csv and .txt format files in the internal private GitHub central repository.[14] To prepare for the blockchain implementation, we preprocessed these SQL codes and summary statistics files collected between May 28, 2020 and June 7, 2021, along with 14 178 user activity logs and 3596 relevant files (ie, SQL codes and data results). The dataset included 27 researcher-provided questions that have been translated into SQL code by the consortium Lead Sites. The institutional-level summary statistics files were encrypted with a hashed value to prevent privacy leaks, while keeping SQL code as a plaintext. Multiple activity logs were associated with the same file when the file was modified, renamed, or deleted after it was first added to the repository. In this case, there were 2 activity logs for one file. The details of data fields captured from each file are summarized and described in Table 1, with several fields being parsed from other fields. Specifically, *Institution* was extracted from the *User* field (35 users in total) and then mapped to the abbreviation names of the 14 institutions (eg, "*UCSD*"). Also, the Source/Target *Query Number* and the *File Type* fields were parsed from the Source/Target *File Name* fields.

Method overview

An overview of our method is depicted in Figure 2. There are 4 components in our method: (1) the data preprocessing step, which filters, parses, and/or cleans up the activity logs and their files; (2) an off-chain log management component that submits the parsed logs/files to the on-chain program of our blockchain network; (3) a smart contract (ie, programs that are stored and executed on the blockchain) and is responsible for the actual on-chain storing/querying of logs and files; and (4) an activity querying component, which allows on-chain searching of stored logs/files. Together, the components facilitate a workflow in which preprocessed activity logs and files would be recorded, and also made available for querying such as, "*Which institutions responded to Query 26 in May 2021, and what are the result files they sent?*" The outputs are logs and files that satisfy the search. The details of these 4 components are introduced in the following 4 sections.

Data preprocessing

The 4 data preprocessing steps are as follows:

1. Filtering and splitting of logs: we removed administrative logs that were not directly related to COVID-19 clinical research activities (eg, dashboard statistics). This step reduced our dataset size from 14 178 to 9433 logs.
2. Filtering and deidentifying of files: similarly, we removed administrative files to only keep the 3 file types of MD (general description of a particular research query), SQL (the actual code that entails programming language-specific details), and CSV (SQL result of institutional-level summary statistics). An example of excluded file types is a human-readable research question in PDF format. We also "deidentified" any personal information in the files. Specifically, we removed the "requester" information in each MD file and replaced the actual content of a CSV result with a random Secure Hash Algorithm 1 (SHA-1) value with a length of 20 bytes.[49] This was for the purpose of synthetic testing; in a real-world deployment, authentic data can be transmitted among authorized network members securely via the SHA-1 algorithm. This step reduced our dataset size from 3596 to 2143 files.
3. Linking logs and files: the file names mentioned in the activity logs were used to identify the associated files. For logs with operation "*R*" (*Rename*), we used the *target* file names for the linkage, because after renaming the *source* file names would have been outdated. This step did not change the size of our dataset.
4. Cleaning-up and parsing logs/files: in this final step, we removed duplicated "typo" logs such as those with simple upper/lower case issues. Then, we also removed the files that were not associated with any activity logs (eg, because of the removal of the duplicated typo logs). Next, we parsed the fields of the logs to derive other necessary information (ie, those with "*" as shown in Table 1). After this step, we reduced the dataset size to 9166 logs and 2098 files, to be recorded and made available for

Table 1. Information captured from the COVID-19 consortium private GitHub repository

Type	Field	Description	Example(s)
Log	Timestamp	Unix timestamp of the operation	*1613693792* (Friday, February 19, 2021, 12:16:32 AM)
	User	User who performed the operation and is identified by their email address	tskuo@ucsd.edu
	Institution*	User's institution, derived and mapped from *User*	*UCSD*
	Operation	Operation conducted	*A* (Add), *D* (Delete), *M* (Modify), *R** (Rename)
	Source File Name	Full name of the source file ("*N/A*," ie, "not applicable," for the "*A*" operation, and the same as the target file name for the "*M*" operation)	*Query_0026/results/Site01_results_new.csv*
	Source File Type*	Type of file, derived from *Source File Name*	*MD* (description of the query), *SQL* (query details of the query), *CSV* (local/aggregated results)
	Source Query Number*	Numerical reference for the COVID-19 queries, derived from *Source File Name*	*0, 1, 2, . . ., 26*
	Target File Name	The full name of the target file ("*N/A*" for the "*D*" operation, and the same as the source file name for the "*M*" operation)	*Query_0026/results/Site01_results.csv*
	Target File Type*	Type of file, derived from Target File Name	*MD*, *SQL*, or *CSV*
	Target Query Number*	Numerical reference for COVID-19 queries, derived from *Target File Name*	*0, 1, 2, . . ., 26*
File	Compressed File Content	Content of the operated file after compression	*Site01_results.csv*

Note: The fields with asterisks (*) were derived from other fields. In the original GitHub log, there is a "similarity score" after the "R" operation (eg, "R98," indicating that the target file, after renaming, has 98% similarity with the source file). In our study we only focused on the operation and considered "R98" as "R".

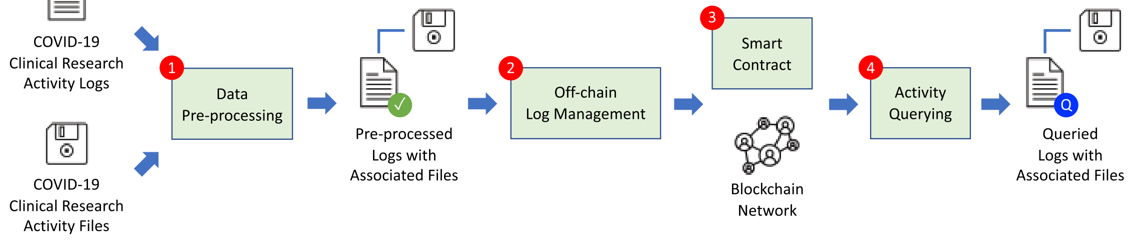

COVID-19 Clinical Research Activity Logs
COVID-19 Clinical Research Activity Files
1
Data Pre-processing
Pre-processed Logs with Associated Files
2
Off-chain Log Management
3
Smart Contract
Blockchain Network
4
Activity Querying
Queried Logs with Associated Files

Figure 2. Method overview. The inputs include COVID-19 queries/results logs and files, and the outputs are the queried logs with associated files. There are 4 components: (1) data preprocessing, (2) off-chain log management, (3) smart contract, and (4) activity querying.

searching on the blockchain. Among the 9166 logs, only 4250 logs were associated with files; this was due to the fact that one file could be linked to multiple activity logs. For example, a "Rename" operation would result in at least 2 activity logs for one single file.

As shown in Figure 3, the filtering process mainly removed non-clinical research activity logs/files, thus the impact to the effectiveness of our dataset is minimal.

Off-chain log management

After activity logs and associated files had been preprocessed, we developed an off-chain log management component to automate the submissions of logs/files to the blockchain network. To make the input data compatible with the smart contract design (detailed in the "Smart contract" section), we parsed out necessary information from the activity logs such as the original timestamp of the research activity, the user email address, the institution to which the user belonged, the operation committed ("add," "modify," "delete," or "rename"), the specific research query that the activity log pertained, and the content of the associated file (if any). Although the COVID-19 clinical research activity logs are relatively small, the files associated with such activities were up to ~330 KB. As a key consideration of storing data on blockchain is scalability,[33] we compressed each file before submitting it to the blockchain network to improve the storage speed. The parsed fields and compressed files were then ready to be passed onto the next component (ie, the on-chain smart contract, to be described in the next section). In addition to automating the submission of logs/files to the blockchain for storage purposes, the off-chain log management component also automated the querying task by submitting search criteria to the blockchain network, that is, it would "call" or "invoke" the on-chain querying function and receive the returned matches (more details in the "Activity querying" section).

Smart contract

We developed a smart contract to store and query the COVID-19 clinical research query/result activity logs with their corresponding files on-chain with 2 main functions, storage and querying, to be invoked by the off-chain log management component (described in the "Off-chain log management" section). For the storage function, each activity log as well as its associated compressed file content

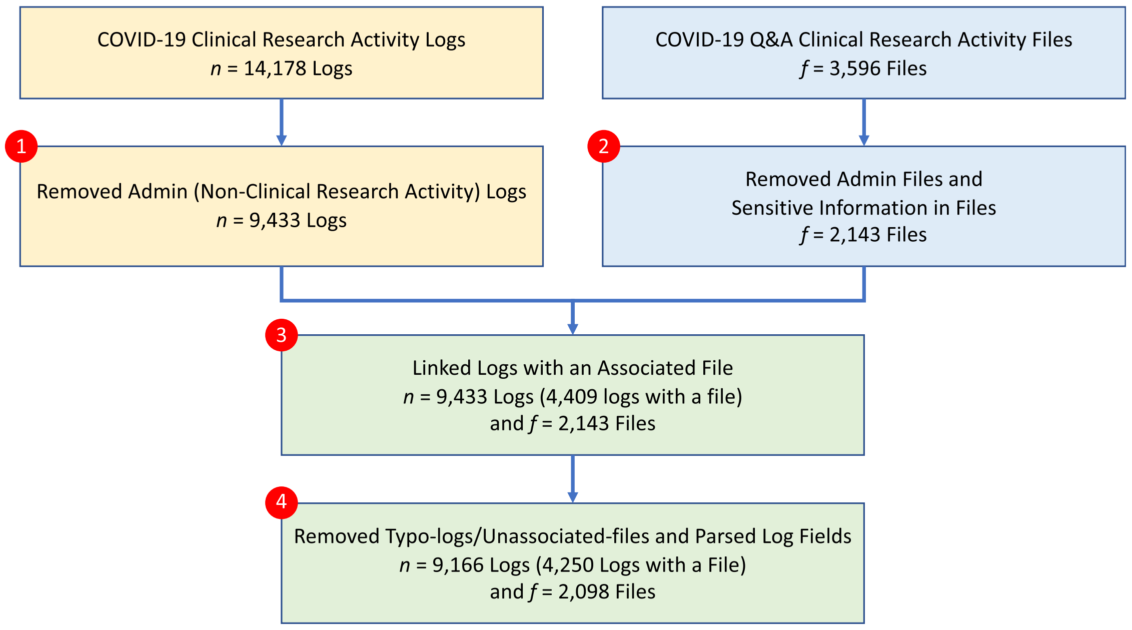
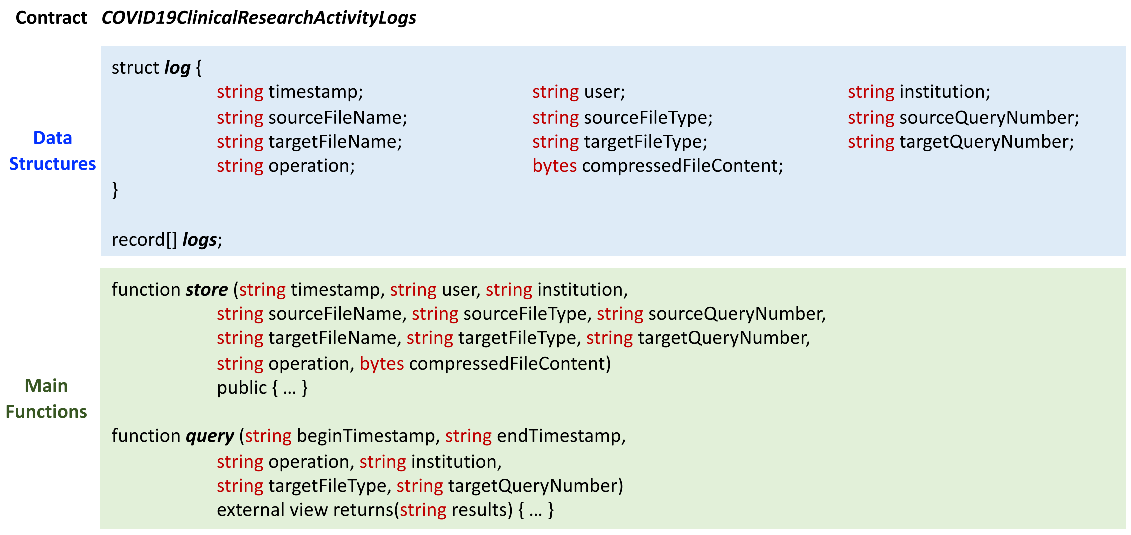

COVID-19 Clinical Research Activity Logs
n = 14,178 Logs
COVID-19 Q&A Clinical Research Activity Files
f = 3,596 Files
1
Removed Admin (Non-Clinical Research Activity) Logs
n = 9,433 Logs
2
Removed Admin Files and Sensitive Information in Files
f = 2,143 Files
3
Linked Logs with an Associated File
n = 9,433 Logs (4,409 logs with a file)
and f = 2,143 Files
4
Removed Typo-logs/Unassociated-files and Parsed Log Fields
n = 9,166 Logs (4,250 Logs with a File)
and f = 2,098 Files

Figure 3. Data preprocessing flowchart. The preprocessing steps include: (1) filtering and splitting logs, (2) filtering and deidentifying files, (3) linking logs and files, and (4) cleaning-up and parsing logs/files.


Data Structures
Main Functions

```
Contract  COVID19ClinicalResearchActivityLogs

struct log {
            string timestamp;            string user;                  string institution;
            string sourceFileName;       string sourceFileType;        string sourceQueryNumber;
            string targetFileName;       string targetFileType;        string targetQueryNumber;
            string operation;            bytes compressedFileContent;
}

record[] logs;

function store (string timestamp, string user, string institution,
            string sourceFileName, string sourceFileType, string sourceQueryNumber,
            string targetFileName, string targetFileType, string targetQueryNumber,
            string operation, bytes compressedFileContent)
            public { … }

function query (string beginTimestamp, string endTimestamp,
            string operation, string institution,
            string targetFileType, string targetQueryNumber)
            external view returns(string results) { … }
```

Figure 4. The high-level smart contract developed to store/query COVID-19 clinical research activity logs. The data structure mainly contains an array of "line of log," and each line of log consists of all fields listed in Table 1 (including the *Compressed File Content* if a file is associated with this line). The store/query main functions are designed for recording and searching the logs on the blockchain.

(if any) is recorded on the blockchain in the form of a "data struct," which is a set of smart contract variables, to store the parsed-out activity information and the compressed file (when appropriate). For the querying function, our smart contract design also allowed user-specified search criteria to filter the variables within the data structs. The high-level data structure and functions are demonstrated in Figure 4, and more specific details about the querying capacity are provided in the next section.

Activity querying

To allow researchers and network auditors to search the COVID-19 clinical research data stored on-chain, we developed the query component with the following 6 search criteria: *Begin Date*, *End Date*, *Operation*, *Institution*, *Query Number*, and *File Type*. Since the target query number and file type are more up-to-date than their source counterparts, we only focused on target fields (eg, Query Number and File Type). The *Begin Date* and the *End Date* were converted to Unix timestamps to be compared with the logs and files' Timestamp fields listed in Table 1. After querying, logs as well as their associated files (if any) that met the search criteria were returned to the researchers/auditors. Both the search criteria and the querying matches were passed between the off-chain log management component and the blockchain smart contract.

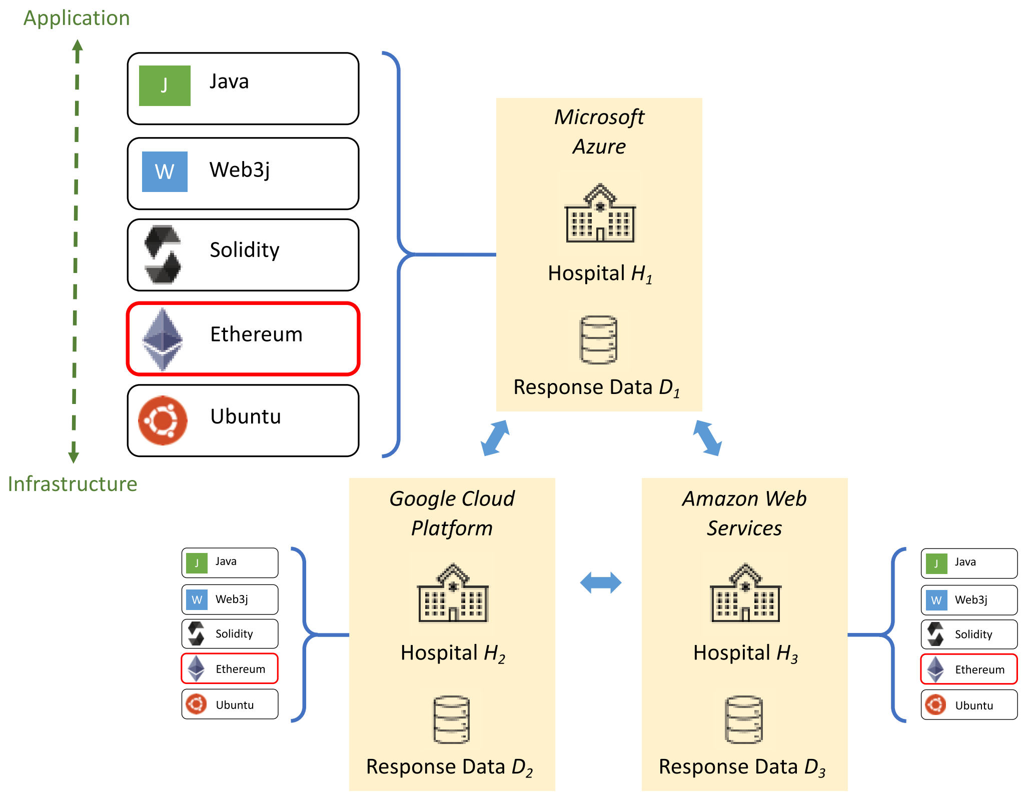


Figure 5. Implementation architecture. We simulate a 3-node blockchain network, with each node hosted in a different cloud provider. The technology stack on each hospital is the same (with upper being application-level and lower infrastructure-level). Ethereum is the only connection point to other nodes.

GitHub repository baseline

To understand the performance of our proposed solution versus a centralized one, we developed a GitHub baseline where users initiate SQL code to the consortium's GitHub repository and await member results via file commits. Since GitHub is designed to store files (instead of data entries), fields of information regarding timestamp, user email, institution, etc., were concatenated into a string, so as the string could be pushed to the repository as a file commit (with compressed file content if the specific activity log included an actual file, empty otherwise). Taking into consideration the inner working mechanism of GitHub, in which a file needs to be sequentially added, committed, and pushed from a local repository that is fully up-to-date with its remote counterpart, we first created multiple branches/subrepositories (one for each node), to which the particular node would push a file without the need for frequent pulling. After all nodes had finished pushing to their individual branches, a coordination script merged the branches into a single, final repository. Then, we queried the file commits using the search criteria (described in the "Activity querying" section) to find matches on a local repository that had pulled all contents from the remote repository (ie, a "pull" action) to ensure the local repository is always up-to-date with the remote, making the "pull" action an inherent part of the query.

Cross-Cloud system implementation

The implementation architecture of our system is shown in Figure 5. We simultaneously deployed our system on 3 major cloud providers: Microsoft Azure (MA), Google Cloud Platform (GCP), and Amazon Web Services (AWS). Based on our prior reviews,[50,51] we selected the community-supported, open-source Ethereum[52] blockchain platform. Specifically, we adopted the Go-Ethereum (Geth) implementation,[53] and used a Proof-of-Authority (PoA)[54] consensus protocol Clique[55] which is designed for private blockchain (ie, only invited institutional nodes/computers may join the network). We developed the smart contract using the Solidity[56] programming language, utilized Java and Bash to develop the off-chain log management and log querying components, and leveraged Web3j[57] as the bridging library between our off-chain Java application and the blockchain network. The baseline methods were implemented using Python and Bash. We deployed our system on Ubuntu Virtual Machines (VMs) with 2 vCPUs, 8GB RAM, and 100GB Disk. This same specification was adopted for all 3 cloud providers. Considering the fact that cloud platforms may be susceptible to privacy leakage, we only uploaded the deidentified files (ie, Step 2 in the "Data preprocessing" section) to the cloud platforms to conduct the experiments to protect privacy.

Experiment settings

To demonstrate its practical feasibility, we evaluated the initialization, data recording, and search time performance of the system, and compared it with the baseline. We simulated 2 scenarios in our experiments, referring to different cases of how the data load could have been distributed: *log-level*, and *institution-level* data splitting. For *log-level* splitting, we randomly split the 9166 logs (and any

Table 2. Possible values per query criterion

Field	Values	Number of combinations
Begin timestamp	6/1/2020, 9/1/2020	2
End timestamp	3/1/2021, 6/1/2021	2
Operation	A, D, M, R	4
Institution	CS, CUA, EMORY, GT, LMU, SMMC, UCD, UCI, UCLA, UCSD, UCSF, USC, UTH, VA	14
Query number	00–26	27
File type	MD, SQL, CSV	3
Total number of combinations		18 144

Note: For *Begin* and *End* timestamps, we selected 2 candidate dates each.

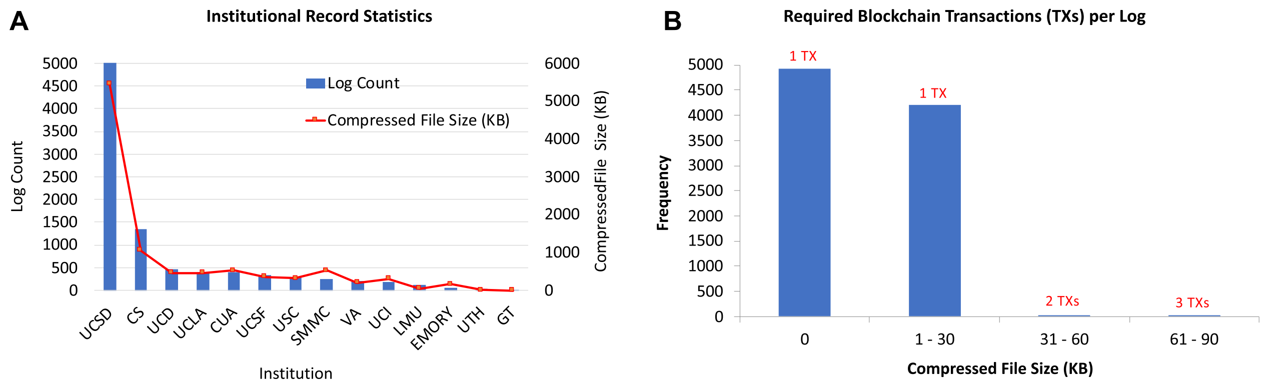


Figure 6. Statistics of records and compressed file sizes. (A) Institutional record statistics. The *X* axis shows the 14 participating institutions, ordered by the record count (left *Y* axis, blue bars). The total compressed file size for each institution is depicted in the right *Y* axis. (B) Blockchain transactions (TXs) per record. Each blockchain transaction has a size limit of ∼30 KB for the Ethereum blockchain platform. Also, the logs without an associated file (ie, compressed file size=0 KB) still required 1 TX to be stored on the blockchain. Therefore, the *X* axis shows the range of the compressed file sizes for 1, 2, and 3 TXs, and the *Y* axis demonstrates the frequency of the records within the corresponding size range.

associated files) into approximately 3 equal parts, which were stored on the 3 VMs, each within a different cloud (ie, MA, GCP, or AWS). This was to simulate the ideal situation in which each blockchain node has approximately an equal number of logs/files in their data inputs. For *institutional-level* splitting, we first split the 9166 logs/files to 14 bins based on their institutions. We then randomly grouped the 14 bins of logs/files into 3 parts to be stored again on 3 VMs within a different cloud. For example, a bin might have 5 institutions, and the other 2 bins would have 4 institutions each. This scenario simulated the more realistic situation where each blockchain node may include a cluster of institutions with vastly different activity loads, and thus each contains a vastly different number of logs/files as inputs.

For both scenarios, we measured the time required for initialization (ie, deploying the smart contract on the blockchain). Next, we measured the average recording time per log, which was measured by the duration between the time the first recording transaction was initiated and the time when the last transaction finished among all VMs. This was particularly important in the *institutional-level* data splitting scenario, as a node might have significantly more input logs than other nodes, and thus would require more time to finish all transactions. We repeated this process 30 times and conducted a paired *t* test with alpha = 0.05 to understand whether there are statistically significant differences ($P < 0.05$) between our proposed method and the baseline.

After the logs and files were recorded on-chain, we conducted 18 144 queries using the exhaustive combinations of 6 query criteria (details in Table 2) and measured the average search time per query. An example of a query combination could be "*Find all activity logs and files within 9/1/2020 and 3/1/2021 that are of operation type 'Modify' ('M'), were initiated by UCSD for Query 9, and were CSV extensions.*" We also compared the results with the ones from the baseline methods.

RESULTS

COVID-19 clinical research data analysis

At the institutional level, the log statistics are shown in Figure 6A. The log count and the compressed file size both follow a long-tailed distribution (ie, most logs/files were from the more actively participating institutions, and most files after compression were of size up to 30 KB).[58] Some institutions may store larger files per log, indicated by the relatively higher compressed size value when compared with its activity log count. The required number of transactions (TXs) per log is shown in Figure 6B; for example, after compression, the largest file became ∼90 KB, which required 3 blockchain TXs with a size limit of ∼30 KB each for the Ethereum blockchain platform used in our experiment. The logs without an associated file still required 1 TX to be recorded on-chain. The compressed file sizes ranged from 0 to 90 KB, and the corresponding numbers of required blockchain TXs ranged from 1 to 3, both following a long-tail distribution. For log-level data splitting, 9166 logs were evenly split into 3 nodes, and the maximum number of logs per node was $\lceil 9166/3 \rceil = 3056$ during any of the 30 trials. For the institution-level data splitting, the number of logs on each node may have differed due to the randomization of institutional combinations; on an average, a

Table 3. Storage time results

Splitting	Measurements	Baseline	Proposed	*P* value
Log-level	Total time	28 811.467 (2571.143)	6405.482 (88.000)	$<10^{-28}$
	Average time over 9166 records	3.143 (0.281)	0.699 (0.010)	$<10^{-28}$
	Average time over the largest number of input logs	9.428 (0.841)	2.096 (0.029)	$<10^{-28}$
Institution-level	Total time	35 769.300 (7864.710)	13 401.609 (1882.121)	$<10^{-14}$
	Average time over 9166 records	3.902 (0.858)	1.462 (0.205)	$<10^{-14}$
	Average time over the largest number of input logs	5.713 (1.183)	2.130 (0.201)	$<10^{-15}$

Note: The time was measured in seconds. The values were averaged over 30 trials, with standard deviation shown in parentheses.

node might carry up to 6783 logs per trial, while the other nodes had fewer input logs.

Smart contract deployment times

The average smart contract deployment time for the log-level data splitting was 4.557 s, which is a one-time cost. The average deployment time for the institution-level splitting was 4.430 s. The standard deviations of log- and institution-level splitting were 0.354 and 0.203 s, respectively.

Storage times

The results for storage times are shown in Table 3. In general, the total time for the baseline and the proposed blockchain method depended on the distribution of input logs among nodes and was in direct proportion with the largest number of input logs among the 3 nodes. However, the storage speeds between the 2 comparing methods showed statistical differences for both data splitting scenarios of the 9166 input logs: For the *log-level* splitting scenario, each node had either 3055 or 3056 input logs, indicating that each VM should finish pushing to the blockchain or GitHub around the same time. The GitHub baseline took ∼8 h with a relatively high standard deviation (∼43 min) over 30 trials. The proposed blockchain network achieved consistent performance (∼1.8 h) with a relatively low standard deviation (∼1.5 min) across 30 trials. For the *institution-level* splitting, the GitHub baseline took an average of ∼10 h, versus the proposed blockchain's ∼3.7 h, averaged over their relative 30 trials each. Their corresponding standard deviations showed a similar trend (ie, the one for the blockchain method is smaller than that of the baseline) to the log-level splitting case.

The differences in total time and consistency between the 2 methods translated to their speed differences when averaged over all 9166 input logs or the largest number of input logs assigned among the 3 VMs during a particular trial. When averaged over the largest number of input logs, the GitHub baseline took ∼9.4 s at log-level and ∼5.7 s at the institution-level on an average. Meanwhile, whether log-level or institution-level, the blockchain took ∼2.1 s on an average.

Query times

For correctness, the on-chain query result from the proposed blockchain mechanism agreed 100% with the returned matches as yielded locally by running a Python querying script on a fully updated GitHub local repository (a "pull" command was run as part of each query). Each local GitHub query took ∼2.4 s for log-level splitting and ∼1.8 s for institution-level splitting. Meanwhile, the blockchain query would take ∼0.5 s for either splitting. Detailed information on querying speed of the proposed blockchain method is illustrated in Table 4.

Table 4. Query time results

Splitting	Measurements	Proposed
Log-level	Total time	10 670.472 (4730.111)
	Per-query time	0.588 (0.261)
Institution-level	Total time	7537.861 (361.600)
	Per-query time	0.415 (0.020)

Note: Time is measured in seconds, and per-query values are averaged over 18 144 combinations and 30 trials, with standard deviation shown in parentheses.

DISCUSSION

Findings

In general, the deployment time (∼4 s) was negligible since the smart contract only needs to be deployed once when the blockchain network is being set up. Compared to the GitHub baseline, our proposed blockchain mechanism was more effective at storing federated data analysis activities among consortium members ($P < 10^{-28}$ in the log-level and $P < 10^{-14}$ for the institution-level splitting scenarios). The difference in speed between the 2 systems might be attributed to the fact that GitHub does require frequent reconciliation of states between local and remote repositories before each commit can be made, and that the sequence of actions entailing "add/commit/push" might also need more time to complete. There might also be external factors that affected GitHub performance such as server traffic.

Similarly, the proposed blockchain was also faster in querying (∼4 times as fast), which demonstrated the feasibility of log auditing. It took advantage of the fact that a blockchain network node always has the most up-to-date data record due to the distributed ledger infrastructure, whereas the GitHub local/remote setting mandates a thorough "pull" as part of each query. The feasibility of our cross-cloud blockchain-based system for recording clinical research query/result activities was thus demonstrated by the low deployment time, activity recording time, and querying time, showing that from a technical perspective, blockchain with its native security edges can function as the supporting infrastructure for communications among members of clinical data consortiums.

We also implemented and tested a *round-robin* design of the GitHub storage method, which in effect most closely resembled the blockchain mechanism by letting all nodes/computers publish to the same remote repository. Specifically, before any node could push a file commit from local to remote, it would need to successfully pull down the most current state of the remote repository (ie, wait for the completion of the previous node, thus "round-robin"). The average storage time of this design is ∼42.4 s for log-level and ∼21.8 s for institution-level splitting, which is slower than our GitHub baseline (∼9.4 s and ∼5.7 s, respectively) as well as our proposed blockchain method (∼2.1 s for both splitting scenarios).

Contribution of the blockchain-based solution

Although GitHub was not specifically designed for federated data analysis nor team communication, we used it as our baseline in this study to be consistent with the R2D2 Consortium's mechanism.[14] By design, GitHub is vulnerable to potential server outages (and thus carries the single-point-of-failure threat). In comparison, our proposed blockchain approach offers both faster storage and querying speeds, and the benefit of being highly available without single-point-of-failure; given that each network node always has a full copy of the blockchain, our mechanism will not be impacted if any of the participating nodes fails. This technical feature emphasizes the benefit that our blockchain-based solution may have over other centralized methods such as GitHub.

Based on results that demonstrated the feasibility of using blockchain for clinical research activity logging, we anticipate our system can be adapted for other collaborative research activities beyond COVID-19. For example, it may be of use in modern times when global transportation increases the risk of spreading diseases (eg, monkeypox, Zika virus, etc.), and the data consortium model is an ideal candidate to gather data from simultaneous, disconnected outbursts. We believe that blockchain-based solutions, while not being "one-size-fits-all," would bring the most value when multiple institutions would like to collaborate for a time-critical mission, and where issues such as single-point-of-failure would impede such collaborations. Nevertheless, it is imperative to weigh the costs and benefits of adopting this new distributed technology in lieu of a mature and centralized one. Moreover, an experienced team that understands how to design and develop blockchain-based solutions in an effective and efficient way will be an important consideration when generalizing this study for different application fields.

Limitations

There are limitations to our study:

1. The format of our queries remained in the SQL format as described in the "Data" section, and thus is different from NLP-based question-answering methods such as CoQUAD[3] that take natural language texts as their input. We are yet to investigate and adopt NLP methods[59,60] to automatically parse the COVID-19 questions from their natural texts.
2. We only evaluated our system on a blockchain network with 3 nodes (one from each cloud provider), while more nodes unevenly distributed across 3 clouds might impact system performance.
3. We assumed that the requests to store/query the COVID-19 research queries and results were spread evenly at any time. However, since multiple users may simultaneously work on one blockchain node, chances are people may suddenly have interest in the same research topics (eg, a new virus variant appears) in a short time.
4. We are yet to redesign our system to accommodate such potential surges of storing/querying requests and compare our system with its centralized counterpart.
5. With regards to the size of files that can be recorded on-chain, we have yet to investigate the limit of the size.
6. Blockchain is a linear linked-list data structure by design. Therefore, the query time may increase as the blockchain grows. Methods to improve querying scalability such as ones reported in recent benchmarking studies[44,45] are yet to be investigated.
7. We are yet to evaluate our method in real-world settings such as when pandemic data may increase exponentially. While the direct inputs for our system are SQL queries and statistical results, which may not scale in synchronization with the abundance of patient data at each institution, the pandemic may still impact the frequency of federated data analysis activities and/or the number of data consortium members. Further work would be needed to investigate both the scalability of the system and the use cases for other pandemics such as monkeypox.

CONCLUSION

Our results show that it is feasible to store and query COVID-19 queries and results on blockchain in the way similar to conventional platforms such as GitHub, to take advantage of blockchain's intrinsic desirable technical features such as decentralization, immutability, and high availability. Given that our experiment utilized 3 different cloud environments, they may reflect a real-world setting, where a data-contributing consortium consists of multiple institutions whose computational architectures are placed on different cloud providers. This study can help support the recording/management of future clinical federated data analysis activity logging and file transferring systems, where data from as many sources as possible are required to yield a better picture of disease progression to inform public health, and especially to better combat pandemics.

FUNDING

The authors were funded by the U.S. National Institutes of Health (NIH) (R01HG011066, R01HL136835, R01GM118609, R01EB031030, R00HG009680, U24LM013755, UL1TR001881, and U54HG012510), a UCSD Academic Senate Research Grant (RG100836), 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.

AUTHOR CONTRIBUTIONS

T-TK contributed to conceptualization, data curation, formal analysis, investigation, methodology, project administration, resources, software, validation, visualization, and writing (original draft). AP contributed to formal analysis, investigation, methodology, software, validation, visualization, and writing (review and editing). ME contributed to formal analysis, investigation, methodology, software, validation, visualization, and writing (review and editing). JK contributed to data curation, formal analysis, investigation, methodology, software, validation, and writing (review and editing). JC and YG contributed to investigation and writing (review and editing). LO-M contributed to conceptualization, funding acquisition, project administration, resources, supervision, and writing (review and editing). The R2D2 Consortium (the full list included in the Supplementary Appendix) contributed to data curation and writing (review and editing).

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

ACKNOWLEDGMENTS

The authors would like to thank Cyd Burrows-Schilling, MS for the technical support for the UCSD Campus AWS cloud network. We thank Daniel Edelson, PhD for his guidance in our implementation tasks. We also thank Robert E. Murphy, Sebastian Niedermayer, William A. Carter, Amy M. Perkins, and Michael Aratow for their support.

CONFLICTS OF INTEREST STATEMENT

Hua Xu has financial interest at Melax Technologies Inc. Other authors declare no competing interests.

DATA AVAILABILITY

The deidentified version of data used in this article will be shared on reasonable request to the corresponding author.

REFERENCES

1. Temesgen ZM, DeSimone DC, Mahmood M, Libertin CR, Palraj BRV, Berbari EF. Health care after the COVID-19 pandemic and the influence of telemedicine. *Mayo Clin Proc* 2020; 95 (9): 66–8.
2. Else H. How a torrent of COVID science changed research publishing – in seven charts. *Nature* 2020; 588 (7839): 553–4.
3. Raza S, Schwartz B, Rosella LC. CoQUAD: a COVID-19 question answering dataset system, facilitating research, benchmarking, and practice. *BMC Bioinformatics* 2022; 23 (1): 1–28.
4. Alzubi JA, Jain R, Singh A, Parwekar P, Gupta M. COBERT: COVID-19 question answering system using BERT. *Arab J Sci Eng* 2021: 1–11.
5. Wu X, Zheng H, Dou Z, *et al.* A novel privacy-preserving federated genome-wide association study framework and its application in identifying potential risk variants in ankylosing spondylitis. *Brief Bioinformatics* 2021; 22 (3): bbaa090.
6. Pezoulas VC, Kalatzis F, Exarchos TP, *et al.* Dealing with open issues and unmet needs in healthcare through ontology matching and federated learning. In: 8th European Medical and Biological Engineering Conference: Proceedings of the EMBEC 2020, November 29–December 3, 2020; Portorož, Slovenia; 2021. Springer.
7. Petkova E, Antman EM, Troxel AB. Pooling data from individual clinical trials in the COVID-19 era. *JAMA* 2020; 324 (6): 543–5.
8. McBride O, Murphy J, Shevlin M, *et al.* Monitoring the psychological, social, and economic impact of the COVID-19 pandemic in the population: context, design and conduct of the longitudinal COVID-19 psychological research consortium (C19PRC) study. *Int J Methods Psychiatr Res* 2021; 30 (1): e1861.
9. Kochunov P, Jahanshad N, Sprooten E, *et al.* Multi-site study of additive genetic effects on fractional anisotropy of cerebral white matter: comparing meta and megaanalytical approaches for data pooling. *Neuroimage* 2014; 95: 136–50.
10. Cao H, McEwen SC, Forsyth JK, *et al.* Toward leveraging human connectomic data in large consortia: generalizability of fMRI-based brain graphs across sites, sessions, and paradigms. *Cereb Cortex* 2019; 29 (3): 1263–79.
11. Drew DA, Nguyen LH, Steves CJ, *et al.*; COPE Consortium. Rapid implementation of mobile technology for real-time epidemiology of COVID-19. *Science* 2020; 368 (6497): 1362–7.
12. Li MM, Pham A, Kuo T-T. Predicting COVID-19 county-level case number trend by combining demographic characteristics and social distancing policies. *J Am Med Inform Assoc Open* 2022; 5 (3): 1–11. doi: 10.1093/jamiaopen/ooac056.
13. Edelson M, Kuo T-T. Generalizable prediction of COVID-19 mortality on worldwide patient data. *J Am Med Inform Assoc Open* 2022; 5 (2): 1–9. doi: 10.1093/jamiaopen/ooac036.
14. Kim J, Neumann L, Paul P, *et al.*; R2D2 Consortium. Privacy-protecting, reliable response data discovery using COVID-19 patient observations. *J Am Med Inform Assoc* 2021; 28 (8): 1765–76.
15. COVID-19ClinicalDataConsult. What is the prevalence of venous thromboembolism (VTE) among the different races of patients with COVID-19? 2021. https://covid19questions.org/component/content/article/32-q-a/88-what-is-the-prevalence-of-venous-thromboembolism-among-the-different-races-of-patients-with-covid-19?Itemid=279. Accessed March 20, 2023.
16. Lardinois F. Microsoft says GitHub now has a $1B ARR, 90M active users. 2022. https://techcrunch.com/2022/10/25/microsoft-says-github-now-has-a-1b-arr-90m-active-users/. Accessed March 20, 2023.
17. Docs G. Access permissions on GitHub. https://docs.github.com/en/get-started/learning-about-github/access-permissions-on-github. Accessed March 20, 2023.
18. McLaughlin K. The Florida COVID-19 data 'whistleblower' crashed the state's dashboard and locked out her manager before she was fired, the National Review reports. 2021. https://www.businessinsider.com/rebekah-jones-florida-covid-19-whistleblower-dashboard-national-review-2021-5. Accessed March 20, 2023.
19. Grind K, McMillan R. Meta employees, security guards fired for hijacking user accounts. 2022. https://www.wsj.com/articles/meta-employees-security-guards-fired-for-hijacking-user-accounts-11668697213. Accessed March 20, 2023.
20. Zhao Q, Huang C, Dai L. VULDEFF: vulnerability detection method based on function fingerprints and code differences. *Knowl Based Syst* 2023; 260: 110139.
21. Marczak B, Scott-Railton J, McKune S, Abdul Razzak B, Deibert R. *Hide and Seek: Tracking NSO Group's Pegasus Spyware to Operations in 45 Countries*. Citizen Lab Research Report No. 113, University of Toronto; 2018.
22. Pallini T. American Airlines and others carriers were left helpless after a system outage crippled operations, causing delays. 2021. https://www.yahoo.com/now/american-airlines-others-carriers-were-183048782.html. Accessed March 20, 2023.
23. Dutton J, Joyner A. United Airlines says system outage resolved after U.S. flights stopped. 2021. https://www.newsweek.com/united-airlines-ground-stop-computer-systems-down-1630138. Accessed March 20, 2023.
24. Colonial Pipeline Cyber Incident. 2022. https://www.energy.gov/ceser/colonial-pipeline-cyber-incident. Accessed March 20, 2023.
25. Strickland E. 5 Major hospital hacks: horror stories from the cybersecurity frontlines. 2016. https://spectrum.ieee.org/5-major-hospital-hacks-horror-stories-from-the-cyber-security-frontlines. Accessed March 20, 2023.
26. Chanthadavong A. Sabre systems IT outage cripples airline operations globally. 2023. https://www.zdnet.com/article/sabre-systems-it-outage-cripples-airline-operations-globally/. Accessed March 20, 2023.
27. Harwell D. New York stock exchange outage adds to fears on financial markets. 2023. https://www.washingtonpost.com/business/economy/nyse-outage-raises-questions-about-whether-regulators-can-keep-up/2015/07/08/e8f7b02a-258d-11e5-b77f-eb13a215f593_story.html. Accessed March 20, 2023.
28. Tsidulko J. United Airlines, NYSE outages reveal poor redundancy architecture, insufficient testing | CRN. https://www.crn.com/news/security/300077385/united-airlines-nyse-outages-reveal-poor-redundancy-architecture-insufficient-testing.htm. Accessed March 20, 2023.
29. Ballinger K. April service disruptions analysis. 2020. https://github.blog/2020-05-22-april-service-disruptions-analysis/. Accessed March 20, 2023.
30. Ballinger K. An update on recent service disruptions. 2022. https://github.blog/2022-03-23-an-update-on-recent-service-disruptions/. Accessed March 20, 2023.

31. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. *Decentralized Bus Rev* 2008: 21260.
32. Hasselgren A, Kralevska K, Gligoroski D, Pedersen SA, Faxvaag A. Blockchain in healthcare and health sciences—a scoping review. *Int J Med Inform* 2020; 134: 104040.
33. 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 (6): 1211–20.
34. Atlassian. git amend | Atlassian Git Tutorial. 2023. https://www.atlassian.com/git/tutorials/rewriting-history. Accessed March 20, 2023.
35. Göbel T, Baier H. Anti-forensics in ext4: on secrecy and usability of timestamp-based data hiding. *Digit Investig* 2018; 24: S111–20.
36. Ekblaw A, Azaria A, Halamka JD, Lippman A. *A Case Study for Blockchain in Healthcare: "MedRec" Prototype for Electronic Health Records and Medical Research Data*. Gaithersburg, MD: ONC/NIST Use of Blockchain for Healthcare and Research Workshop; 2016.
37. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: using blockchain for medical data access and permission management. In: International Conference on Open and Big Data (OBD); 2016; Vienna, Austria, IEEE.
38. Kuo T-T, Pham A. Quorum-based model learning on a blockchain hierarchical clinical research network using smart contracts. *Int J Med Inform* 2023; 169: 104924.
39. Kuo T-T, Pham A. Detecting model misconducts in decentralized healthcare federated learning. *Int J Med Inform* 2021; 158: 104658.
40. Kuo T-T, Kim J, Gabriel RA. Privacy-preserving model learning on blockchain network-of-networks. *J Am Med Inform Assoc* 2020; 27 (3): 343–54.
41. Kuo T-T, Gabriel RA, Cidambi KR, Ohno-Machado L. EXpectation Propagation LOgistic REgRession on permissioned blockCHAIN (ExplorerChain): decentralized online healthcare/genomics predictive model learning. *J Am Med Inform Assoc* 2020; 27 (5): 747–56.
42. Kuo T-T. The anatomy of a distributed predictive modeling framework: online learning, blockchain network, and consensus algorithm. *J Am Med Inform Assoc Open* 2020; 3 (2): 201–8.
43. Kuo T-T, Gabriel RA, Ohno-Machado L. Fair compute loads enabled by blockchain: sharing models by alternating client and server roles. *J Am Med Inform Assoc* 2019; 26 (5): 392–403.
44. 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 Genomics* 2020; 13 (Suppl 7): 98.
45. 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.
46. 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.
47. Tellew J, Kuo T-T. CertificateChain: decentralized healthcare training certificate management system using blockchain and smart contracts. *J Am Med Inform Assoc Open* 2022; 5 (1): 1–9. doi: 10.1093/jamiaopen/ooac019.
48. 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 (12): 2182–90.
49. Stevens M, Bursztein E, Karpman P, Albertini A, Markov Y. The first collision for full SHA-1. In: Annual International Cryptology Conference; 2017; Springer.
50. 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 (5): 462–78.
51. 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.
52. Buterin V. A next-generation smart contract and decentralized application platform. *White Paper* 2014; 3 (27): 2–1.
53. TheEthereumCommunity. Go Ethereum (Geth). https://github.com/ethereum/go-ethereum. Accessed March 20, 2023.
54. Wood G. Proof-of-Authority (PoA) Private Chains. 2015. https://github.com/ethereum/guide/blob/master/poa.md. Accessed March 20, 2023.
55. TheEthereumCommunity. Clique PoA protocol & Rinkeby PoA testnet. 2017. https://github.com/ethereum/EIPs/issues/225. Accessed March 20, 2023.
56. TheEthereumCommunity. The Solidity Contract-Oriented Programming Language. 2022. https://github.com/ethereum/solidity. Accessed March 20, 2023.
57. Web3Labs. web3j: Web3 Java Ethereum Dapp API. https://github.com/web3j/web3j. Accessed March 20, 2023.
58. Liu Z, Miao Z, Zhan X, Wang J, Gong B, Yu SX. Large-scale long-tailed recognition in an open world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019.
59. Badia A. Question answering and database querying: bridging the gap with generalized quantification. *J Appl Log* 2007; 5 (1): 3–19.
60. Hirschman L, Gaizauskas R. Natural language question answering: the view from here. *Nat Lang Eng* 2001; 7 (4): 275–300.