JAMIA Open, 2024, **7(2)**, ooae040
https://doi.org/10.1093/jamiaopen/ooae040
Research and Applications

Research and Applications

An infrastructure for secure data sharing: a clinical data implementation

Joanna F. DeFranco, PhD*[,1], Joshua Roberts, BS[2], David Ferraiolo, BS[2], D. Chris Compton, MS[2]

[1]Computer and Information Science, Penn State University, Malvern, PA 19355, United States, [2]National Institute of Standards and Technology, Secure Systems and Applications Group, Gaithersburg, MD 20899, United States

*Corresponding author: Joanna F. DeFranco, PhD, Computer and Information Science, Penn State University, 30 East Swedesford Road, Malvern, PA 19355, United States (jfd104@psu.edu)

Abstract

Objective: To address database interoperability challenges to improve collaboration among disparate organizations.

Materials and Methods: We developed a lightweight system to allow broad but well-controlled data sharing while preserving local data protection policies. We used 2 NIST-developed technologies—Next-generation Database Access Control (NDAC) and the Data Block Matrix (DBM)—to create a proof-of-concept system called the Secure Federated Data Sharing System (SFDS). NDAC controls access to database resources down to the field level based on attributes assigned to users. The DBM manages and shares authoritative user-attribute assignments across a federation of organizations, implemented using a modified open-source permissioned blockchain, to manage and share authoritative user-attribute assignments across a federation of organizations. We used synthetic data to demonstrate a clinical research data-sharing use case using the SFDS.

Results: We demonstrated, through consent, the onboarding of previously unknown users into NDAC via assignments to their DBM-validated attributes, allowing those users policy-preserving access to local database resources. The SFDS main system components—NDAC and DBM—also showed excellent performance metrics.

Discussion: The SFDS provides a generic data-sharing infrastructure that effectively and securely achieves data-sharing objectives. It is completely transparent to the otherwise normal business operations of participating organizations. It requires no changes to database management systems or existing methods of authenticating and authorizing local user access to local resources.

Conclusion: This efficiency, flexibility of deployment, and granularity of control make this new infrastructure solution practical for meeting the data-sharing and protection objectives of the clinical research community.

Lay Summary

The ability to share database resources among collaborating organizations is highly desirable. However, sharing data continues to be a challenge when it comes from different types of database management systems (DBMSs) due to different schemas and data formats. Another challenge that is limiting collaboration is being able to enforce the home organizations local data protection policies while sharing data. For example, can a patient's condition details be shared with an outside research organization—for a limited time—without sharing the patient's sensitive diagnosis? Can it be shared without data integrity concerns? Some of the issues are not related to the technology but to the trust of the system. Trust is integral when allowing someone to directly query your database. Current industry attempts have challenges addressing these data sharing problems. The current solutions are a hodgepodge of approaches and do not effectively and securely achieve all data-sharing objectives. The Secure Federated Data Sharing System (SFDS) was designed to be more comprehensive and transparent to the otherwise normal business operations of participating organizations. It is a practical solution in that it requires no changes to an organizations DBMS, nor existing methods of authenticating and authorizing local user access to local resources. The SFDS authenticates external users by the validating mandatory attributes and ultimately assigning discretionary attributes to access local data. Using a clinical research use case, this paper demonstrates and describes the SFDS technology to facilitate secure data sharing among disparate organizations while enforcing local data access/protection policies at a granular level—without disruption of normal business operations. In other words, this paper describes how data can be shared with outside organizations while leaving the data at the home organization—rather than exchanging or being centrally stored. The SFDS is an effective, lightweight solution to a pervasive database resource sharing problem.

Key words: distributed ledgers; data sharing; access control; privacy; security.

Introduction

The ability to share database resources among collaborating organizations is highly desirable. This is especially true in the performance of clinical research and operations, where data sharing can enhance patient care outcomes, reduce operational costs for hospitals, and potentially improve efficiency for clinical trials and pharmaceutical development data management.

Historically, collaboration in this space has been curtailed by concerns about sharing highly regulated and sensitive health data among disparate medical institutions due to database interoperability, incompatibility, inadequate access controls, and the potential to inadvertently disclose protected patient information.[1] Other concerns include fear of unauthorized disclosure of sensitive data that involves ongoing

Received: October 10, 2023; **Revised:** February 2, 2024; **Editorial Decision:** April 22, 2024; **Accepted:** April 23, 2024
Published by Oxford University Press on behalf of the American Medical Informatics Association 2024.
This work is written by (a) US Government employee(s) and is in the public domain in the US.

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

scientific innovation or discovery, and other intellectual property concerns. With differing risk tolerance, cultures, resources, and expertise, healthcare institutions naturally take on different protection postures. For instance, Institution A may have a very large dataset with very few restrictions on sharing with Institution B, which has significant research expertise and resources. In contrast, Institution C may have a more restrictive posture with regard to sharing due to the nature of the population it serves.

The sharing of database management systems (DBMSs) data is also technically hard, regardless of its type. This is because the data is often in different formats and schemas and comes from different systems, making it difficult to transfer, consume, and correctly interpret among organizations. Existing approaches vary dramatically, are costly to implement, are often inefficient, and are intrusive to operations. For example, clinical trial data can be used beyond its original purpose to analyze new hypotheses, avoid research duplication, and identify subpopulation effects, but the challenge is expensive and requires resource-intensive solutions to share data.[2] The need for security compounds the problem. DBMS data is often sensitive for varying reasons and can only be accessed by users in accordance with host organization policies. The existing technologies used to enforce such policies can also unintentionally restrict data access to the point where secure access means limiting the use of other software and the merging of data from other sources.[2]

This paper describes the Secure Federated Data Sharing (SFDS) system, which is designed to address these challenges. While supportive of a range of industry use cases as described in DeFranco et al.[3] This paper is an extension of DeFranco et al[3] by describing the proof-of-concept implementation of the SFDS in detail. In this implementation, the focus is on the application of the SFDS to promote collaboration in support of clinical research. NIST recognizes that the secondary usage of electronic health record data for research has increased dramatically in the last decade, resulting in discoveries that improve the well-being of individuals and support the decision-making process for medical stakeholders (eg, physicians, clinicians, nurses, and administrators). Our objective is to make it easier to facilitate broad but well-controlled secondary access to medical data by experts. The approach standardizes policy-preserving access to data where it currently resides rather than its exchange or centralized storage. An important virtue is that the data-sharing infrastructure is non-intrusive and completely transparent to the otherwise normal business operations of participating organizations. This ease of deployment, granular access, and efficiency make SFDS practical for meeting the data-sharing and protection objectives of the clinical research and operations community.

Background

Since structured data is highly searchable via human-generated queries and algorithms, DBMSs are and will remain a primary means for storing and managing healthcare data for the foreseeable future. To meet this demand, a wide variety of database products are available, each providing a powerful means for extracting, creating, and updating data in accordance with the broadly recognized SQL standard.

Given the sensitivity of much of the data, controlled access in accordance with policy is a fundamental requirement. While policy makers have specified a large variety of access controls to address real-world security issues, only uneven subsets of these policies can be enforced in database scenarios. Furthermore, today's approach to access control is by no means standardized, requiring separate configurations of an often complicated amalgamation of mechanisms, including those that are custom implemented in applications and specific to DBMS products.

In general, ensuring conformance to access rules among multiple organizations makes authorized access to transmitted data difficult to govern and audit. That is, the policy-preserving retrieval of data can be a heavy lift when considering the possibility of patient data being lost or misused when stored on a third-party remote site.[4] Permitting access to the data where it resides avoids these issues but is also problematic due to the need to grant local access to previously unknown users with unknown credentials to databases with varying authorization mechanisms.

1) In the healthcare space, the dominant approach of sharing data is through conformances to the Fast Healthcare Interoperability Resources (FHIR) specification for the exchange of healthcare records. FHIR was promoted and has been adopted to solve interoperability issues among healthcare systems through a common schema for transferring records to other systems to facilitate clinical care. The issue with clinical research, however, is sharing datasets and not records with protection policies that go beyond the Health Insurance Portability and Accountability Act (HIPAA) requirements. Gaps remain with FHIR regarding the control of access between collaborating institutions and the ability to query records for exploratory data analysis at a broad scale.[5] There are also adoption challenges related to heavyweight implementation costs (eg, significant investment in previous standards, no toolkit to enable system architecture, etc.).

As central repositories, research data warehouses integrate research and patient data from 1 or more sources into a single data model that is designed for research, using the process of loading data from source systems, called Extract, Transform, and Load (ETL). The i2b2 Research Data Warehouse was designed to streamline the ETL process. It allows refresh as opposed to full reload, which can not only be conducted more frequently, but at the granularity of the datasets that have changed and need updating. SFDS provides a standardized means of policy-preserving access to the data where it currently resides, rather than its exchange, or centralized storage, thus eliminating the need for transfer or loading data in the first place.

From an access control perspective, research data warehouses do not make a distinction between the access capabilities of its users, but rather different warehouses, deploy different models in support of different research topics for their user populations. SFDS takes the perspective that data can only be accessed by users in accordance with the policies of the host organization. These policies may emanate from concerns of inadvertent disclosure of protected patient information, as might a data warehouses, but also concerns of unauthorized disclosure of other sensitive data such as that involving ongoing scientific innovation or discovery, and other intellectual property concerns. Sharing of such data can

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

vary based on user credentials and pertain to organizational affiliations.

Regarding controlling access to the database, itself, there has been an effort to reduce data redundancy with data segregation (eg, creating subject oriented data marts). This solution utilizes PostgreSQL's row level security (RLS) feature to minimize replication and synchronization—where managing security roles to implement data marts is proposed.[6] This solution is effective; however, each implementation needs to be configured for different database (DB) formats (eg, Postgress, Oracle, etc.). As mentioned, the SFDS uses a standard approach that is DB agnostic and provides access control down to the granularity of a single cell.

Solution: SFDS technology overview and system operation

SFDS applies an Attribute Based Access Control (ABAC) model to grant or deny access to resources based on user and resource assignments to attributes and a set of policies that are specified in terms of those attributes. SFDS solves database-sharing problems while allowing policy-preserving access to resources stored in the databases of organizations that belong to a federation. Through the automatic onboarding of users and their federation-validated attributes in a DBMS-agnostic authorization system, users are provided policy-preserving access to data where it currently resides. This solution is based on the integration of 2 patented and proven NIST technologies: Next-generation Database Access Control (NDAC) and the Data Block Matrix (DBM).[7]

NDAC provides a universal access control layer between applications and DBMS resources as a middleware that translates a user's query to a permitted (policy-preserving) query that is submitted to a database.[8] NDAC is DBMS-agnostic and can enforce access policies at a granularity that is not usually available to DBMS products and accommodates the protection concerns of institutions willing to share health data. In keeping with the ABAC model, NDAC controls access to DBMS resources based on the attributes assigned to its users. The DBM is a type of distributed ledger that provides a means for storing, managing, and sharing the attributes of users in the federation. Through access consent (with potential restrictions), previously unknown users are onboarded into NDAC systems via assignments to their DBM-validated attributes, thus allowing them policy-preserving access to database resources. The exchange of attributes is fundamentally the means for allowing access to DBMS resources, and the DBM instills trust in those attributes.

Implementations of Next Generation Access Control (NGAC), an ANCI/INCITS standard (https://webstore.ansi.org/standards/incits/incits5652020), supports both NDAC and the DBM. The expression of policy in terms of NGAC's access control data provides the basis for query translation in its support of NDAC. A different NGAC implementation provides a means of controlling administrative actions as to who and under what circumstance can create and delete user-to-attributes assignments as reflected in the DBM. In effect, this later use of NGAC serves SFDS as an access control intermediator between attribute administrators and the DBM.

Next-generation database access control

NDAC provides a universal access control layer between applications and DBMSs by leveraging NGAC's underlying ABAC model.[9]

As illustrated in Figure 1, ABAC is an access control model in which a user's request to access resources is granted or denied based on the assigned attributes of the user, the assigned attributes of the resource, and a set of policies that are specified in terms of those attributes. ABAC is highly flexible in its specification of a large variety of enterprise-specific access control policies. Assigning a user to 1 or more attributes provides that user with resource access rights that are consistent with the policies, and assigning a resource to attributes grants access rights to that resource in a manner that is consistent with the policies.

NDAC is DBMS-agnostic, does not require modification of the DBMS software, and can enforce access policies at a granularity that is not usually available to DBMS products. In keeping with the ABAC model, NDAC controls access to DBMS data based on attributes assigned to the users as a middleware that sits between applications and the DBMSs. Consequentially, NDAC eliminates the need to utilize access controls that are implemented in conventional applications and/or in instances of proprietary DBMS mechanisms. This approach simplifies and standardizes the means of protecting databases from queries sent from any application with control down to the field level.

NDAC uses NGAC policy elements and relations as a means of expressing policies to include attributes. NGAC treats attributes as containers for characterizing its user and object members. For object-attributes, this is ideal for representing complex data structures, such as relational database tables with distinguished fields. The rows of a table may be expressed as containers of objects that correspond to the row's fields, and columns may be expressed as containers of objects that correspond to column fields.

The NDAC process for expressing access control policies begins with an existing DBMS schema, which includes columns and tables that are automatically converted into NGAC-corresponding object attributes and assignments. Given that rows are also object containers, existing rows could be automatically converted as well.

NGAC data elements and relations also include a generic set of access rights, and 3 types of relations for specifying an access policy. Once the DBMS schema has been converted, NGAC relations are configured in formulating policy in terms of the created object attributes and assignments using NGAC's administrative API. The resulting data elements and relations are stored as NGAC Access Control Data.

Thanks to linear-time algorithms for reviewing NGAC access control data, NDAC can efficiently "see" DBMS resources (ie, rows, columns, tables, and fields) that a user is authorized to access in advance of the user issuing a query. This not only provides a valuable means of conducting a "before-the-fact audit" but also provides the basis for interpreting SQL statements. That is, NDAC can generate a permitted SQL statement in the case of Select or *approve* or *deny* access in the case of Insert, Delete, or Update based on a user's known access capabilities.

In addition to the NGAC access control data store, NDAC includes an Access Manager to trap SQL queries from applications, a Translator to convert SQL queries, and a user

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

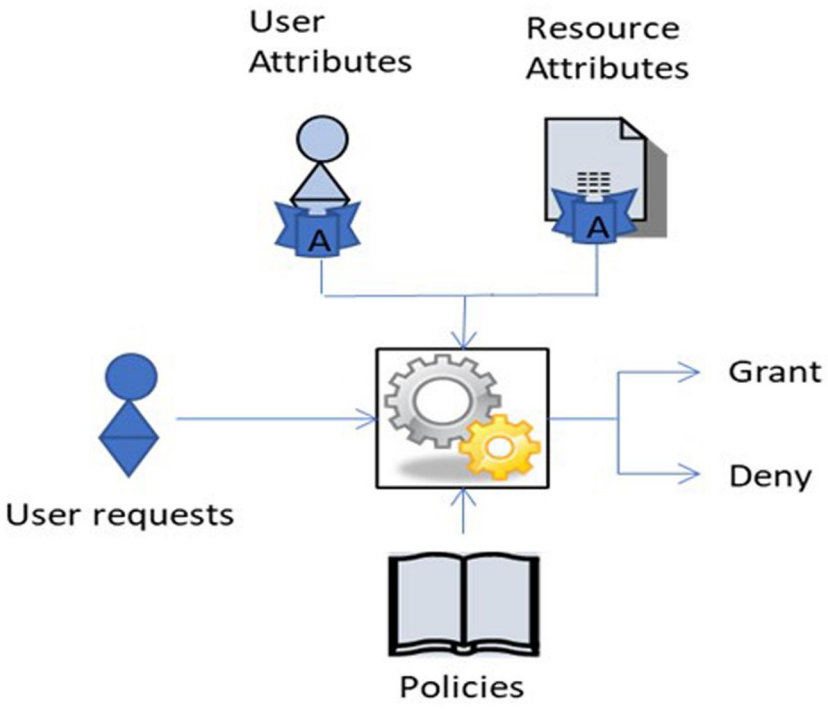

Figure 1. Attribute based access control (ABAC).

identity for permitted SQL queries. Thus, users may issue a query to fetch an entire data set, and NDAC will restrict access to the subset of data permissible for the user in accordance with local regulatory requirements and consent expressed in terms of combinations of Discretionary Access Control (DAC) and Mandatory Access Control (MAC) policies.

DAC enables users to grant or prohibit other users access to resources that are placed under their control. As the name implies, access is established at the discretion of the granter and is often enforced based on the name (an attribute) of the user being granted access. On the other hand, MAC policies impose non-discretionary access rules on users and are often used to enforce regulatory policies or mandated enterprise policies that are unavoidable. Access rules are based on roles or other types of attributes, such as organizational affiliation. Finally, NGAC can enforce policies in combination. In our use case, this pertains to DAC in support of consent and MAC in support of mandated enterprise policies.

Figure 2 shows the placement of NDAC's Access Manager, Translator, and NGAC Access Control Data in an authorization flow that involves an application and a target database. The authorization flow is as follows:

1) The SQL statement from a user of the Application is intercepted by the Access Manager and sent to the Translator.
2) The Translator converts the SQL statement from the user into NGAC Access Control Data queries.
3) Using the results of the queries, the Translator converts the user's SQL statement into either an access DENY or SQL Statements that are permitted for the user and sent back to the Access Manager.
4) In the case of a DENY, the Access Manager forwards the indication to the application user. Otherwise, the Access Manager submits the Permitted SQL Statement to the database.
5) In the case of a Select operation, data extracted from the database is sent back to the Access Manager and forwarded to the Application and user.

Depending on the type of query (ie, Select, Update, Insert, or Delete), the Translator issues different queries to the NGAC Access Control Database. The results depend on the attributes that are assigned to the user who issues those queries. NDAC's performance is efficient. For example, a query rewrite takes approximately 3 seconds per 1 million rows.

Given NDAC's underlying ABAC model, users gain access to DBMS resources by virtue of attribute assignments.

Data block matrix

The DBM is a new type of distributed ledger with the hashed data integrity protection of a blockchain, but with the additional ability to edit or delete data. It provides a means for storing, managing, and sharing attributes of users in the federation (ie, a form of personal identifiable information [PII]).

This type of user attribute storage is ideal for SFDS because these assignments need to be altered over time and can accommodate privacy regulations, such as the deletion of PII when it is no longer needed.

The DBM (Figure 3) uses an array of blocks with hash values for each row and column. When a cell is deleted or modified, only the hash value for that column and the hash value for that row are altered. The integrity of the other blocks is ensured by the hash values of their rows and columns.

In the SFDS system, the DBM is used to globally manage and share the attribute assignments (eg, user u1 has attribute "physician") across the federation. It is important to note that the attribute names are standardized across the

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

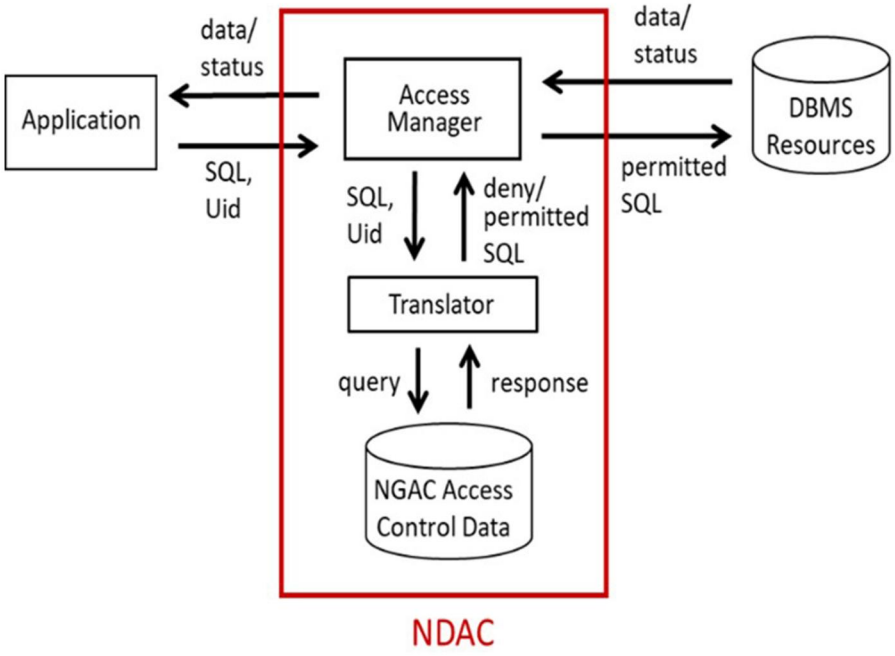

Figure 2. Placement of NDAC (in red) with respect to existing components.

	0	1	2	3	4	
0	•	1	3	7	13	$H_{0,-}$
1	2	•	5	9	15	$H_{1,-}$
2	4	6	•	11	17	$H_{2,-}$
3	8	10	12	•	19	$H_{3,-}$
4	14	16	18	20	•	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

Figure 3. The data block matrix—editing or deleting block 12 only requires the update of row hash $H_{-,3}$, and column hash $H_{-,2}$.

federation, but user assignments to those attributes vary and change over time.

Hyperledger Fabric (HF) was identified as the best solution to implement the features of the DBM because of its highly configurable and modular architecture "enabling innovation, versatility, optimization for a broad range of industry use cases."[10] Furthermore, the structure of blocks in HF and the mechanism in which they are stored allowed for the implementation of the DBM operations while limiting the scope of the modifications. The data in HF blocks consist of bundles of transactions, each with a set of key-value pairs. The keys are the target of the HF read, write, and delete operations. In order to support the DBM delete operation, the write operation is modified to log the blocks a key is updated in. When a key is deleted, each block that referenced the key is updated to remove the value assigned to the key. The block update process consists of deserializing the block's transactions, removing the target key, re-serializing the transactions, re-signing the block, and updating the hashes of the row and column which contain the block's number. Blocks are updated sequentially in order to validate the row and column hashes after each update. The end result will be a set of row and column hashes that validate the data in the blocks as if the deleted key never existed.[11]

While there is an ongoing effort to measure the performance impact of the DBM operations, there are no formal measurements yet. However, it is possible to estimate the performance delta of the DBM operations in relation to their original HF counterparts. The read operation is directly comparable to original HF, the only difference is the DBM read operation reads from a key-value database instead of a file. The write operation in the DBM has the added overhead of logging the keys present in the block's transactions. The delete operation is expected to have the largest performance impact because of the need to update each block that references the key being deleted.

System operation

The SFDS system operation begins with governance to establish privileges for organizations to read from and write to the DBM. This provides the basis for trust in the integrity of user attribute assignments during operations (ie, when issuing consent and access to resources). In the SFDS integration, trust is further bolstered by imposing an administrative policy regarding who can create or delete those assignments and under what authority. The source of these administrative rules should come from a mutually agreed-upon governance policy that—in our implementation—is enforced by NIST's NGAC open-source reference implementation, which was described earlier in this paper. A reference implementation can also be found online (https://github.com/PM-Master/policy-machine-core). The governance operation description is beyond the scope of this paper and can be found in a previous article.[12]

Once the governance is implemented and the system is operational, the onboarding process is possible. Onboarding involves adding previously unknown users into the NDAC system—including assignments to their DBM validated attributes—to allow them policy-preserving access to local

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

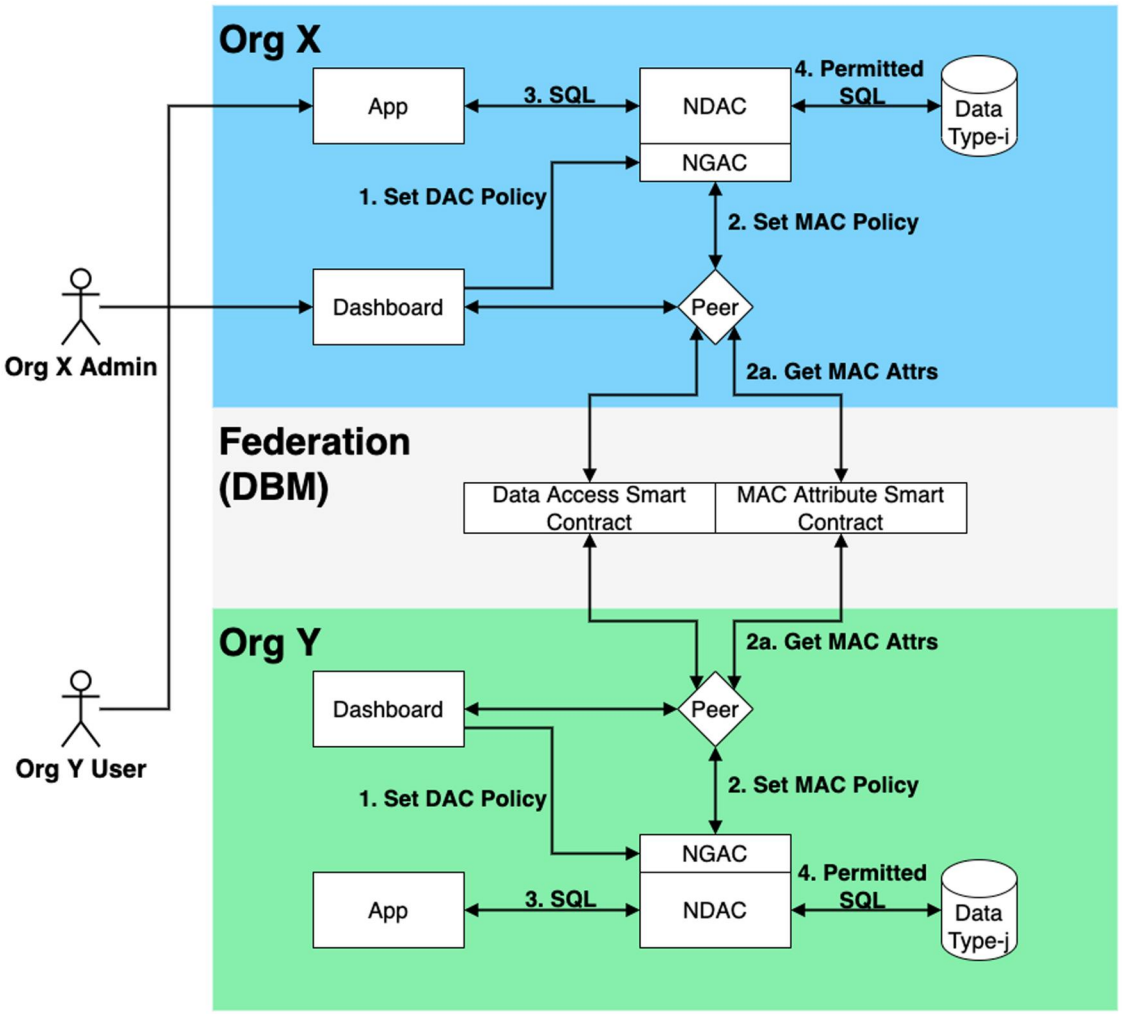

Figure 4. General SFDS system operation.

database resources. These attributes are the means for allowing access, and the DBM instills trust in them.

Operationally (see Figure 4), there are 2 types of end users: those who issue consent and those who access database resources (both types of users are authenticated through a common off-the-shelf identity provider). Consent is an affirmative response to a remote user's (eg, u1 of organization-X) request to access local resources (eg, at organization-Y).

In our implementation, consent is established through NDAC-authorized executions of NGAC standard administrative operations (2) that onboard the requesting user into the local NDAC system by imposing DAC (ie, need-to-know with potential restrictions) and MAC policies on the user. This process includes (1) the administrator at Org X setting the DAC policy for the Org Y user, (2) the MAC policy is set for the Org Y user using the attributes retrieved from the DBM, (3) the Org Y user sends a SQL query, (4) the SQL query is translated to permitted SQL and submitted to the database where data is returned (or denied) to the Org Y user.

Proof of concept: clinical research

The proof-of-concept implementation described in this section is based on collaborative data sharing for clinical research and uses realistic medical records generated using the open-source Synthea synthetic electronic health records software package.[13] The goal is to make it easier to facilitate broad but well-controlled secondary access to the data by experts.

Figure 5 illustrates the high-level scenario in which Researcher 1 at an academic research institution is studying the variables and indicators of patients at risk of sepsis. Researcher 1 requests data from Hospital A and Hospital B. The data analysis from Researcher 1's study will support more accurate predictions of sepsis risk among patients in order to encourage early intervention in these vulnerable populations.

The specific data of interest at both institutions could include admission type, length of stay, mortality, age, gender, comorbidities, vital signs, labs results, and other data. Additionally, the patient data protection policy needs to be supported.

Figure 6 illustrates the operational sharing details of the disparate database resources between Researcher 1, Hospital A, and Hospital B. The process of onboarding Researcher 1 at each institution after consent is granted. "App-A" and "App B" enable data request and consent from the researcher. The "Dashboard" is where the hospital administrator can onboard a researcher and make policy changes,

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

depending on consented access. The sequence of steps are the following: (1) the hospital administrators set their own DAC policy for the researcher, (2) the MAC policy is set for the researcher using the attributes retrieved from the DBM, (3) the researcher sends a SQL query, (4) the SQL query is translated to permitted SQL and submitted to the database where data is returned (or denied) to the researcher.

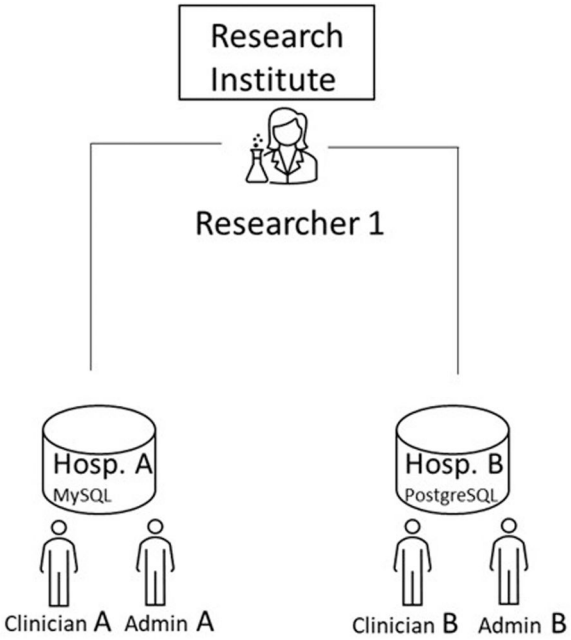

Figure 5. Use case: clinical research collaboration.

The next figures (located in supplementary material: Figure S7 through S14) show screenshots of the SFDS proof-of-concept implementation. Figure S7 shows "Admin A" setting the DAC policy at Hospital A (the MAC policy was previously set at an organizational level).

Figure S8 shows "Clinician A" selecting or requesting specific patient records at Hospital A. The data returned complies with the previously set MAC and DAC policies (Figure S7). The {Empty} field indicates data fields that are hidden due to the set DAC policy.

Figure S9 shows "Researcher 1" selecting or requesting specific patient records at Hospital A. The data returned complies with the previously set MAC and DAC policies (Figure S7). The {Empty} field indicates data fields that are hidden due to the set DAC policy.

Figure S10 shows "Clinician A" selecting or requesting patient procedures at Hospital A. The data returned complies with the previously set MAC and DAC policies (Figure S7). The {Empty} field indicates data that is hidden due to DAC policy (ie, the user does not have access to that field) or the actual value in the table is null.

Figure S11 shows "Researcher 1" selecting or requesting patient procedures at Hospital A. The data returned complies with the previously set MAC and DAC policies (Figure S7). The {Empty} field indicates data that is hidden due to DAC policy (ie, the user does not have access to that field) or the actual value in the table is null.

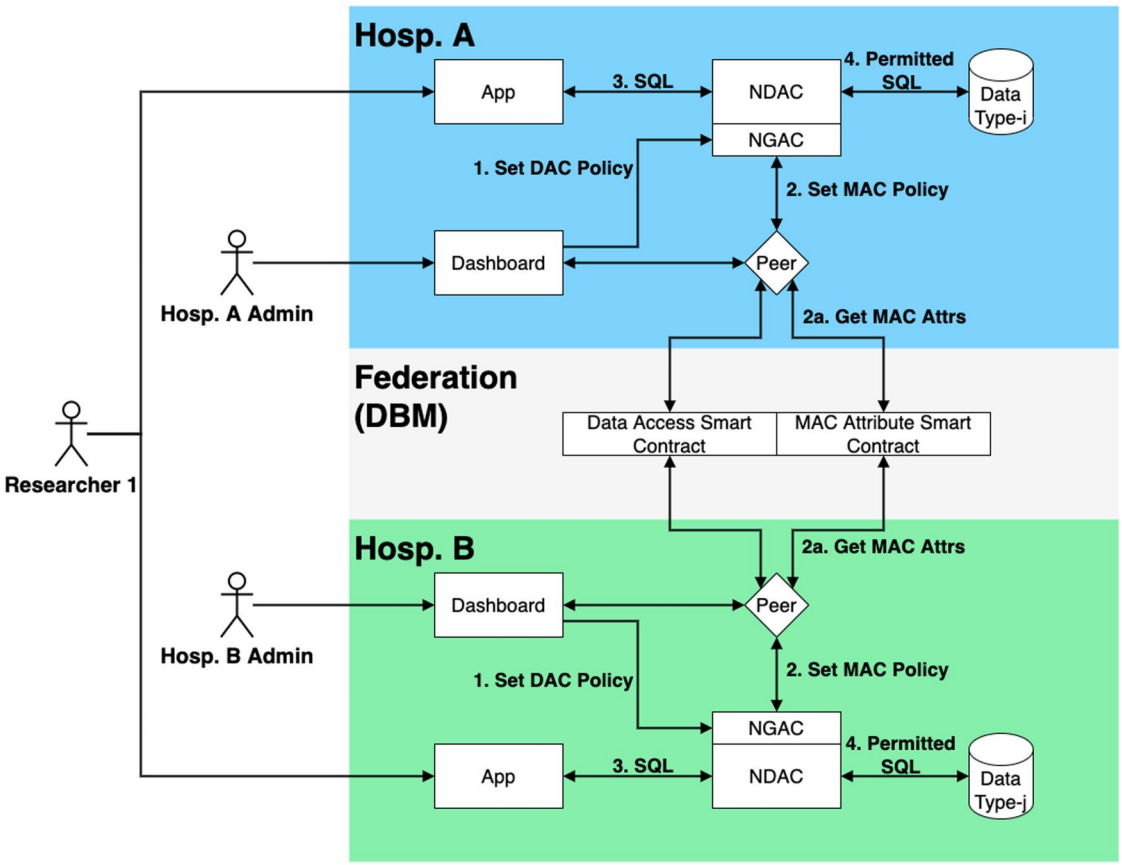

Figure 6. Clinical research collaboration: operational sharing of DBMS resources.

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

Figure S12 shows "Clinician B" selecting treatment data from Hospital B. The MAC policy was previously set at an organizational level and assume the DAC policy was similarly set at Hospital B, as shown in Figure S7 for Hospital A.

Figure S13 shows "Researcher 1," who recall is outside of both Hospital A and B, is selecting treatment data from Hospital B. The MAC policy was previously set at an organizational level, and assume the DAC policy was similarly set at Hospital B, as shown in Figure S7 for Hospital A.

Figure S14 shows "Researcher 1" selecting treatment data from Hospital B after access is revoked (a DAC setting).

Discussion

The SFDS facilitates access to information without moving large volumes of data and with sufficient confidence that the information shared is what was intended. This paper described the SFDS architecture and a proof-of-concept implementation for efficiently sharing clinical data stored in disparate databases.

The SFDS system extends the range of application for fine-grained data sharing beyond what is possible today and effectively creates new markets by bringing valuable commercial and societal impact by dramatically enhancing collaboration. Its flexibility of deployment, scalability, and transparency to otherwise normal business practices make this new infrastructure solution practical for these and numerous other real-world applications. The collaboration challenges addressed by the SFDS are not unique to the healthcare domain. An overview of other possible uses cases, where efficient data sharing is a challenge due to disparate databases, is described in.[11] For example, data sharing between disparate manufacturers can be effective in improving collaboration among stakeholders (supply chain, vendors, customers) as well improving the overall efficiency and productivity of the manufacturing process. Another domain that has had data sharing challenges is law enforcement. An XML framework called the National Information Exchange Model (NIEM) was created to essentially normalize data naming conventions to assist in effectively exchange law enforcement data, however, the NIEM model does not specify how to transmit or store data, making it a partial solution to the law enforcement data sharing problem.[3] The SFDS could be an effective solution to share law enforcement data among multiple states.

Future work and conclusion

The ability to share database resources among collaborating organizations is highly desirable. However, challenges persist regarding interoperability in the exchange of resources between organizations and the preservation of local access policies.

The SFDS provides a generic data-sharing infrastructure that effectively and securely achieves data-sharing objectives. It is completely transparent to the otherwise normal business operations of participating organizations. It requires no changes to DBMSs or existing methods of authenticating and authorizing local user access to local resources. This efficiency, flexibility of deployment, and granularity of control make this new infrastructure solution practical for meeting the data-sharing and protection objectives of the clinical research community.

The current focus of SFDS is to allow the sharing of database resources. The SFDS infrastructure will be extended to allow controlled access to non-structured data, such as files. Adoption of the SFDS will begin with a full-scale pilot study using real medical data. This will involve institutions that house medical information and/or conduct clinical research will be organized with the goal of transitioning SFDS from a research project to operational use. We are also working on a playbook for an organization to be able to: deploy, configure, execute and experiment with the SFDS system. The play book will facilitate the user's evaluation of the SFDS with their organization's current architecture and governance structures. A cooperative research and development agreement (CRADA) could be explored with interested organizations (https://www.nist.gov/tpo/partnerships/cooperative-research-and-development-agreement-crada).

Acknowledgments

Compton was affiliated with the National Institute of Standards and Technology at the time of his contributions to this work and is currently affiliated with the Health Resources and Services Administration.

Author contributions

Joanna F. DeFranco, Joshua Roberts, David Ferraiolo, D. Chris Compton contributed to the requirements, design, and architecture of the SFDS. Joanna F. DeFranco led the SFDS use case design and development and writing of this manuscript. Joshua Roberts led the proof-of-concept design and implementation. David Ferraiolo led the system requirements gathering and architecture of NGAC, NDAC, and their integration into the SFDS. D. Chris Compton advised and provided essential healthcare informatics expertise on the use case and comparative analysis. All authors edited and provided critical feedback on all aspects of this manuscript.

Supplementary material

Supplementary material is available at *JAMIA Open* online.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of interest

The authors have no competing interests to declare.

Data availability

The data used in the SFDS proof-of-concept implementation (and underlying this article) was generating using an open-package that provides synthetic patient data (https://github.com/synthetichealth/synthea). The DBM attribute data was pulled from an opensource application that provides medical terminology called SNOMED (https://browser.ihtsdotools.org/?perspective=full&conceptId1=404684003&edition=MAIN/SNOMEDCT-US/2023-09-01&release=&languages=en).

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

NIST disclaimer

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

References

1. Devriendt T, Borry P, Shabani M. Factors that influence data sharing through data sharing platforms: a qualitative study on the views and experiences of cohort holders and platform developers. *PLoS One*. 2021;16(7):e0254202. https://doi.org/10.1371/journal.pone.0254202
2. Kochhar S, Knoppers B, Gamble C, Chant A, Koplan J, Humphreys GS. Clinical trial data sharing: here's the challenge. *BMJ Open*. 2019;9(8):e032334. https://doi.org/10.1136/bmjopen-2019-032334
3. DeFranco JF, Ferraiolo D, Roberts J, Kuhn DR. Use cases for secure and trusted granular data sharing among disparate databases. *IEEE Rel Mag*. 2024;1(1):20-29. https://doi.org/10.1109/MRL.2024.3352570
4. Grimes J, Szul P, Metke-Jimenez A, Lawley M, Loi K. Pathling: analytics on FHIR. *J Biomed Semantics*. 2022;13(1):23. https://doi.org/10.1186/s13326-022-00277-1
5. Agapito G, Cannataro M. An overview on the challenges and limitations using cloud computing in healthcare corporations. *BDCC*. 2023;7(2):68. https://doi.org/10.3390/bdcc7020068
6. Scheible R, Thomczyk F, Blum M, et al. Integrating row level security in i2b2: segregation of medical records into data marts without data replication and synchronization. *JAMIA Open*. 2023;6(3):ooad068. https://doi.org/10.1093/jamiaopen/ooad068
7. Kuhn R, Yaga D, Voas J. Rethinking distributed ledger technology. *Computer*. 2019;52(2):68-72.
8. Ferraiolo DF, Gavrila S, Katwala G, Roberts J. Imposing fine-grain next generation access control over database queries. In: *Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control (ABAC '17)*. Association for Computing Machinery; 2017:9-15. https://doi.org/10.1145/3041048.3041050
9. Hu V, Ferraiolo D, Kuhn R, et al. 2014. NIST special publication 800-162, Guide to attribute-based access control (ABAC) definition and considerations, SP 800-162, Guide to ABAC Definition and Considerations | CSRC. Accessed May 8, 2024. https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-162.pdf
10. Hyperledger Fabric. 2020-2023. Hyperledger-fabricdocs: introduction. Accessed January 1, 2024. https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html#hyperledger-fabric
11. Roberts JD, DeFranco JF, Kuhn DR. Data block matrix and hyperledger implementation: extending distributed ledger technology for privacy requirements. *Distrib Ledger Technol*. 2023;2(2):1. Article 16 (April 2023), 1. https://doi.org/10.1145/3585539
12. DeFranco JF, Ferraiolo D, Kuhn R, Roberts J. A trusted federated system to share granular data among disparate database resources. *IEEE Computer*. 2021;54(3):55-62.
13. Walonoski J, Kramer M, Nichols J, et al. Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *J Am Med Inform Assoc*. 2018;25(3):230-238. https://doi.org/10.1093/jamia/ocx079

Downloaded from https://academic.oup.com/jamiaopen/article/7/2/ooae040/7674880 by National Institutes of Standards (NIST) user on 01 July 2024

Published by Oxford University Press on behalf of the American Medical Informatics Association 2024.
This work is written by (a) US Government employee(s) and is in the public domain in the US.
JAMIA Open, 2024, 7, 1–9
https://doi.org/10.1093/jamiaopen/ooae040
Research and Applications