IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2024, accepted 10 July 2024, date of publication 19 July 2024, date of current version 29 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3431292

RESEARCH ARTICLE

A Blockchain-Based Hybrid Architecture for Auditable Consent Management

OZGU CAN[1], (Member, IEEE), TUNAHAN DAG[2], AND MURAT KANTARCIOGLU[3], (Fellow, IEEE)

[1]Department of Computer Engineering, Ege University, Bornova, 35100 Izmir, Türkiye
[2]Graduate School of Natural and Applied Sciences, Ege University, Bornova, 35100 Izmir, Türkiye
[3]Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA

Corresponding author: Ozgu Can (ozgu.can@ege.edu.tr)

The work of Murat Kantarcioglu was supported in part by NSF under Award OAC-2115094, and in part by the National Center for Transportation Cybersecurity and Resiliency (TraCR) and Cisco Inc.

ABSTRACT Consent management has become an important issue with the increased usage of the Internet and also smart devices that collect personal data. Each country enacts its regulations and laws for consent management. These laws ensure that personal data is not collected without the individual's consent and cannot be processed with a purpose other than the stated purpose. The General Data Protection Regulation (GDPR) has strict rules regarding collecting and processing personal data. This paper proposes a new approach for auditable hybrid consent management systems using blockchain technology and a purpose tree. The suggested approach includes (1) the implementation of a GDPR-compliant consent management system using blockchain and purpose tree; (2) the implementation of an audit mechanism that detects consent violations and corrects consents; and (3) the use of both on-chain and off-chain technologies. The audit mechanism proposed in this paper detects possible violations by performing inspections on every transaction in the system. Besides, it immediately informs the data subject and the competent authorities regarding the relevant violations. As part of this study, a prototype of the architecture is developed as a proof of concept to evaluate the performance of critical components. The obtained experimental results show that the proposed hybrid architecture that use purpose tree effectively supports consent sharing between the parties.

INDEX TERMS Accountability, auditability, blockchain, consent management, GDPR, privacy, purpose.

I. INTRODUCTION

Data is the smallest unit of information that is used as a basis for any kind of calculation. Every day increasing amount of data is being collected with the increased usage of the Internet. Hence, data can be in many forms and it can be interpreted or used for various purposes. This collected data also includes personal data.

The General Data Protection Regulation (GDPR) is a privacy law that was announced by European Union (EU) in 2018. GDPR offers rights and protections to individuals concerning their personal data and aims to establish and protect the fundamental privacy rights of individuals. According to GDPR, personal data is any information relating to an identified or identifiable natural person [1]. Hence, personal data can be basic identification information such as name, address, and date of birth, as well as individual's contact details, financial reports, biometric data, health data, personal preferences, and employment information.

Since personal data includes a wide variety of information related to an individual, it must be handled cautiously. As stated in the GDPR [1], the protection of natural persons regarding the processing of personal data is a fundamental right. Besides, personal data breaches can harm both individuals and organizations [2]. Furthermore, the increasing amount of personal data collected raises profound issues regarding privacy, security, and data ownership [3]. For this purpose, GDPR requires companies to use private data only with the consent of the individuals whose data are being gathered. Thus, GDPR allows data subjects to take complete

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Ni.

control of their personal data such as how and why their data is used. For this purpose, the entity that collects and uses personal data must obtain individuals' consent before using their personal data.

Consent refers to the statement of the individual about how her personal data is accessed, stored, managed, and shared [4]. The GDPR states that for consent to be informed and specific, the data subject must at least be notified about how her personal data will be used and the purpose of the processing operations, such as marketing, research, or medical treatment [5]. Besides, each processing purpose is associated with one or more processing activities that define how personal data is processed, such as recording, storing, or disseminating data [6]. GDPR requests organizations not to use personal data without the individual's consent. Moreover, it states that data subjects have eight fundamental rights and the right to withdraw consent at any time. Therefore, consent management is one of the essential processes in preserving individual privacy and enforcing the ever-evolving privacy laws.

GDPR and similar privacy laws provide more control to users over their data. In this context, it is crucial to collect user's consent periodically [7]. For any system to be GDPR-compliant, a framework for consent management must be built. Consent management is a set of policies that allows determining unambiguous communication of data processing purposes, obtaining consent from individuals, providing options for consent revocation, maintaining proper documentation of consent records, or managing consent purposes. Therefore, data subjects can use this consent management framework to manage their consents.

The goal of this study is to propose an effective solution that handles consent management transparently and securely. Therefore, all actions must be accountable and any privacy violation must be detected. For this purpose, the proposed solution uses blockchain technology.

Blockchain is a decentralized and transparent digital ledger that securely records and verifies transactions across multiple participants [8]. The blockchain technology standardizes the management of trusted information by allowing access and use of consented data while maintaining data protection [7]. Therefore, blockchain addresses the requirements of an effective consent management solution by improving security, enhancing trust, providing transparency, and adding traceability to the system. However, there are some limitations in using blockchain for consent management systems [7].

The main limitation is that the blockchain does not offer the same level of performance and scalability as traditional databases. Besides, blockchain does not comply with GDPR's ''right to erasure'' also known as the ''right to be forgotten''. The GDPR defined the ''right to be forgotten'' as ''*the data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay*'' in Art.17 [9]. The right to be forgotten comprises the removal of all personal data collected on the data subject and all activity related to the personal data. In the blockchain, any block mined on the ledger cannot be modified afterward. Another challenge related to GDPR-compliant blockchain systems is that the GDPR recommends having at least one data controller. The data controller manages the consent of data owners and it must be reachable by the data owners. Therefore, it is important to distinguish the processing activities per processing purpose and the data controller should also explore whether there is an obligation to maintain a record of processing activities [10]. Further, the append-only nature of the blockchain is also a challenge. Therefore, once data is written to the blockchain, it cannot be altered or deleted. Thereupon, modification and deletion operations should be handled carefully to prevent data breaches. Meanwhile, addressing data protection issues at the software design stage instead of adding a burdensome layer of legal compliance to the final system is accepted as the most appropriate approach for privacy engineering [11].

The existing solutions in the literature focus on the probable applications of blockchain technology to manage users' consent and comply with GDPR. Also, the existing solutions propose standalone applications that organizations need to implement the necessary integration modules to perform on their platform. Hence, these solutions are time-consuming and costly. Therefore, there is a need for an effective solution that acts as a middleware between organizations and data owners without requiring organizations to make changes in their existing architecture. This study proposes a solution that addresses these limitations in the existing approaches.

In addition, the proposed solution considers data subjects' purposes in the consent management process. Storing consent for different purposes increases the system and user experience costs since the data subject must manage each purpose. Therefore, an effective solution is also required for the purpose management process of the consent management system.

Consequently, the proposed study offers a holistic solution for dynamic consent management by considering the data subject's purposes and utilizing both on-chain and off-chain technologies to be GDPR-compliant and support the ''right to be forgotten''. In this context, the proposed study mainly focuses on GDPR's *Chapter 3 - Rights of the Data Subject*. Hence, the main articles of interest are as follows:

- Article 7: Conditions for consent
- Article 12: Transparent information, communication and modalities for the exercise of the rights of the data subject
- Article 13: Information to be provided where personal data are collected from the data subject
- Article 14: Information to be provided where personal data have not been obtained from the data subject
- Article 15: Right of access by the data subject
- Article 16: Right to rectification
- Article 17: Right to erasure/Right to be forgotten
- Article 18: Right to restriction of processing

- Article 19: Notification obligation regarding rectification or erasure of personal data or restriction of processing
- Article 20: Right to data portability
- Article 21: Right to object
- Article 22: Automated individual decision-making, including profiling

Therefore, the proposed blockchain-based solution to consent management provides confidentiality, integrity, access control, transparency, rectification, and erasure. Besides, the purpose-based approach of the proposed solution allows portability, restriction of processing and automated processing by determining users' purposes accordingly.

Considering the existing literature, the main focus of this study is:

- Implementing a hybrid consent management system to improve privacy and comply with GDPR.
- Using data subject's purposes to enable querying consented users for the given purpose.
- Implementing an auditable consent management system to detect consent violations.
- Enhancing the performance and scalability issues for effective consent management.

A. CONTRIBUTIONS

This study proposes a novel hybrid GDPR-compliant dynamic consent management system to address the above-mentioned issues. The proposed system uses data subject's purposes and blockchain technology. In the literature, to the best of our knowledge, there is no implementation of a hybrid GDPR-compliant auditable consent management system with a purpose tree. Most of the studies in the field focus on the potential of blockchain technology and its integration into consent management to achieve GDPR compliance.

In this study, we optimized the usage of blockchain for auditing GDPR-compliant data. The proposed solution can be used in every domain where privacy preservation is required. In this context, the medical domain is chosen as the application domain of the proposed solution and the evaluation results are presented.

The innovative contributions of the proposed study are as follows:

i. *Implementing a GDPR-compliant consent management system powered by blockchain technology and purpose tree*: The proposed study integrates data subjects' purposes into the consent management system to effectively manage users' consents by considering their purposes. The personal data must not be processed without the user's consent. An efficient consent management system should enable the user to clearly state her purpose and define different consents for different purposes. Therefore, the proposed study introduces an effective solution by integrating the purpose management process into the consent management system. To the best of our knowledge, there exists no study that integrates data subjects' purposes into a blockchain-based consent management system.

ii. *Combining the transparency, immutability, and decentralization of on-chain technology with the scalability and throughput-boosting capabilities of off-chain technology*: The study proposes a hybrid architecture that is based on both the blockchain technology and database approach. Thus, the proposed solution provides a balance between security, transparency, performance, scalability, and GDPR compliance.

iii. *Implementing an auditable system to detect any consent violations*: The proposed solution involves quick audit, full audit, user audit, log audit, and off-chain audit mechanisms to detect consent violations.

iv. *Experimental analysis:* The proposed solution also aims conducting improvements to achieve the optimum performance results for the consent management. Thus, we optimized the time and cost of blockchain for auditing GDPR-compliant data.

In addition to the mentioned contributions, the proposed study also aims to provide a guideline for future implementations of accountable GDPR-compliant hybrid consent management systems.

The remainder of the paper is structured as follows: Section II focuses on consent management systems, blockchain, and GDPR to provide necessary background information and examines the related work in the field. Section III presents the related work and summarizes the differences between the proposed work and existing studies. Section IV presents the overall architecture and explains each component of the system. Section V explains example violation scenarios and how the proposed system handles them. Section VI presents the implementation and illustrates the flows of the proposed system. Section VII discusses the experimental results. Section VIII evaluates the findings and impacts of the study. Finally, Section IX concludes the study and outlines future works.

II. BACKGROUND

Consent management is essential for preserving privacy and protecting personal data. The GDPR holds organizations accountable for obtaining data subjects' consent for specific purposes such as storing, processing, and sharing their sensitive information. Thus, individuals must provide their consent and understand the issued consent unambiguously and thoroughly. Besides, gathering and managing consent can be challenging for entities that handle large amounts of personal data. These entities need to develop a straightforward, transparent, and understandable process for all parties involved in the consent management process to meet GDPR requirements.

Blockchain, a well-known example of a distributed ledger technology, fulfills the requirements of consent management. Blockchain assures safe and transparent data storage. In consent management, blockchain technology can be used to

provide a clear and unmodifiable log for consent. Thus, data subjects are empowered to manage their consent effortlessly, and organizations are provided with a secure and verifiable trail of consent. It could also facilitate adherence to other GDPR mandates, such as providing data subjects with the right to access and modify their data. Hence, blockchain allows parties to maintain a secure and verifiable log of these transactions.

In brief, consent management with blockchain offers organizations a transparent, secure, and verifiable method for managing personal data and consent. Therefore, this allows organizations to comply with privacy laws such as GDPR and offers data subjects the necessary control and safety over their data. The following sub-sections present the background information on consent management, GDPR, and blockchain technology.

A. CONSENT MANAGEMENT

Consent is an individual's decision about how her personal data is accessed, managed, and shared [4]. GDPR defines 'consent' of the data subject as *any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her* [1]. Moreover, GDPR states that consent Thus, consent is firmly tied to a specific purpose of processing, which is the intended use of an individual's personal information for activities such as medical research, marketing, or data analytics [6]. Also, each processing purpose is linked with one or more processing activities that outline how personal data is handled, such as data recording, storing, or disseminating.

Consent management is a system, process, or set of policies that allow individuals to determine what personal information they are willing to permit data processors to access [12]. Thus, consent management handles consent requirements. For this purpose, consent management must offer the techniques and structures to confirm that people are entirely informed and voluntarily consented to specific purposes, such as consent for marketing, participation in healthcare research, drug development, or financial studies.

The consent management system must also be transparent to make consent management efficient for all parties. For this purpose, consent and purposes should be undoubtedly coupled. Besides, the consent management system should provide complete and precise information to data subjects about the decisions' implications, potential risks, or benefits. The system also obtains a commitment from data subjects to affirm they have given their consent freely.

Today, personal data is gathered, used, and shared in various fields for different purposes. For instance, online service providers frequently get their users' consent for their terms of service and privacy agreements before they can access certain services. In the healthcare sector, healthcare providers comprehensively inform patients about their treatment choices, potential side effects, and the risks and benefits of each choice. Therefore, patients have the right to modify or revoke their consent at any desired time. Consent management allows organizations to obtain permission when handling sensitive data, ensuring compliance with relevant privacy regulations, and preserving individual privacy. Moreover, consent management empowers individuals with control over their personal data. Thus, consent is a crucial issue in data privacy and consent management has gained increasing importance with the rise of technology.

B. GDPR

GDPR is the European Data Protection Regulation that harmonizes data privacy laws across Europe and is applicable as of May 25th, 2018. GDPR significantly increases individuals' control over their personal data. In addition, the GDPR applies to any entity or organization that processes personal data as part of the operations of one of its branches located in the EU, regardless of where the data is processed; or any organization that is established outside the EU and provides services or monitors the behavior of individuals in the EU [13]. For this purpose, GDPR defines the main actors and their key roles in data protection as data controller, data processor, data subject, data protection officer, supervisory authorities, and European Data Protection Board (EDPB). The GDPR grants data subjects several rights to their data, such as information rights, access rights, rectification rights, erasure rights, restriction of processing rights, data portability rights, objection rights, and automated decision-making and profiling rights.

GDPR enforces that the processing of personal data must be transparent, equitable, and lawful. Thus, organizations should be clear and transparent on personal data collection, usage, and sharing practices to be GDPR-compliant. Besides, organizations should also possess a legal justification for such actions. Hence, GDPR imposes obligations on organizations processing personal data as data protection by design and default, data protection impact assessment, data breach notification, data protection officer, and international data transfer. Further, non-compliance with GDPR can result in hefty fines of up to 4% of an organization's annual global turnover or €20 million, whichever is greater [13]. Beyond monetary penalties, organizations might also confront non-financial repercussions, such as reputation damage and the erosion of customer trust. In essence, the GDPR assures a significant advancement in personal data protection and individual rights, establishing a rigorous standard for data protection expected to be adhered to globally.

C. BLOCKCHAIN

Blockchain is a decentralized and distributed technology that allows participants to record and validate transactions. It eliminates the requirement for central authorities and intermediaries. Blockchain is a network of computation systems called nodes, working together to verify and log transactions

on the digital ledger. Each transaction is included in a block, which joins the preceding blocks forming the blockchain. Each block carries a unique hash pointing to the previous block, safeguarding the chain's integrity and security [8].

The essential characteristic of blockchain is its decentralization, transparency and inalterability. All the transactions that exist within a blockchain are documented publicly and transparently. Therefore, once a transaction is recorded it cannot be modified or deleted. Thus, it guarantees that no single peer can tamper with the ledger or perform unethical activities. The main advantages of blockchain are as follows:

- *Decentralization:* Its decentralized and distributed nature removes the requirement for a central authority or intermediary, improving efficiency and reducing fraud or corruption risks.
- *Security:* Blockchain secures transactions and guards against alterations by utilizing advanced cryptographic techniques, providing a reliable means to store and transfer data.
- *Transparency:* Blockchain enhances trust and responsibility by recording all transactions publicly and transparently.
- *Efficiency:* Blockchain optimizes processes, and decreases costs and delays associated with conventional systems by removing intermediaries.

Blockchain architecture can be designed as public, private, consortium, and hybrid:

- A *public blockchain* is a decentralized and transparent ledger of transactions that is open to anyone to participate, validate, and record data. Anyone may join the network, read the complete transaction history, and be a part of the consensus procedure. Public blockchains often utilize a consensus method, such as Proof of Work (PoW) or Proof of Stake (PoS), to obtain consensus among the nodes.
- *Permissioned blockchains*, commonly referred to as *private blockchains*, are blockchain networks with restricted access to a small number of users or organizations. Private blockchains need authorization to access and engage with the network, in contrast to public blockchains, where anybody may join and participate. Compared to public blockchains, private blockchains can achieve better transaction throughput and faster consensus since they only have a small number of participants [14]. The benefit of scalability is due to the less participation in the consensus process.
- *Consortium blockchains* are a type of blockchain that is governed and operated by a consortium or a group of organizations working together. Consortium blockchains represent a fusion of public and private blockchains with a set of organizations collaborating to verify transactions [15].
- *Hybrid blockchains* combine the characteristics of public and private blockchains to create a flexible and secure system [16]. In a hybrid blockchain, certain components operate as public blockchains, where transactions and data are visible and accessible to all participants. These public components ensure transparency, immutability, and decentralized consensus. Certain components of a hybrid blockchain function as private blockchains, restricting access to authorized participants. These private components provide enhanced privacy, control over data access, and faster transaction processing

Blockchain technology is rapidly being integrated into numerous industries and already has a significant impact on some, such as finance, supply chain management, and healthcare. For instance, in the financial sector, blockchain reduces the cost of cross-border payments, increases transparency, and minimizes fraud risks in areas such as trade finance. In the supply chain industry, blockchain enables parties to track goods' movement, thereby improving transparency and efficiency of the process. In the healthcare domain, blockchain offers a secure and transparent method for storing, sharing, and processing medical data. Furthermore, blockchain enhances voting systems' transparency and accountability and provides a more secure and transparent method to track intellectual property ownership and transfer. Thereupon, blockchain technology has revolutionized several industries and played a significant role in the industrial revolution [17].

III. RELATED WORKS

Consent management is a process that manages access to personal data according to the permissions granted by the individual [4]. Thus, it allows individuals and companies to manage and track personal data in a straightforward and user-centric manner [18]. Consent management systems have become a trending topic in the literature after the GDPR directives. The challenge of managing personal data dynamically under GDPR regulations is addressed in [19]. The study presents a consent management system that uses smart contracts and blockchain technology to provide individuals the ability to dynamically manage their consent preferences.

A user-centric solution is proposed in [20] to allow data subjects to manage their consents related to data access by keeping dataset profiles in the form of blockchain consent. The authors propose a blockchain-based solution to enhance transparency, security, and user control over their personal data. For this purpose, a public blockchain network is utilized to enable individuals to grant or revoke consent for their data to be collected and processed by IoT devices. Also, smart contracts enforce and ensure compliance with data protection laws. However, the study lacks an audit mechanism.

Similarly, smart contracts are also used in [21] to create an individual consent model for health data sharing. The study emphasizes the use of smart contract to enforce consent conditions and automate the consent management process. Thus, data requesters can search and access data using smart contracts. Still, the study does not provide a purpose tree.

In [22], a purpose-based consent model is proposed to allow patients in the medical domain to specify the purpose of data usage and provide granular consent accordingly. The proposed solution focuses on patient data sharing in hospitals, and data donation for biobank research purposes. Nevertheless, the study lacks GDPR compliance, implementation, and performance evaluation. Similarly, the study presented in [23] focus on the usability of blockchain in the healthcare domain and integration of patient consent in the data sharing operations. However, the proposed solution does not include the GDPR compliance. Alike, the development of a blockchain ecosystem for the medical domain is presented in [24]. The study focuses on addressing the challenges of a dynamic consent management system for the medical domain in the context of data sharing and research. The study lacks GDPR compliance, audit mechanism, and the performance evaluation of the prototype implementation. The challenges of consent management in the healthcare domain are also addressed in [25]. The study proposes a framework that leverages blockchain technology to enhance the transparency, security, and efficiency of e-health consent management. The study uses a purpose tree to allow users to define access control. Nevertheless, the study does not address the dynamic consent management and performance evolution. Another blockchain-based solution is presented in [26]. The proposed solution aims to protect health records. However, the study only focuses on the healthcare domain. Besides, the auditing and user purposes aspects are not included in the proposed solution. Also, the presented evaluation is limited to the storage and sharing phases.

In [27] the challenges of consent management in clinical trial research are studied. The authors propose a system based on private blockchain technology to enhance transparency, security, and control over participant consent in dynamic clinical trial settings. The study presents a prototype implementation of the system; however, it does not provide a performance evaluation.

A blockchain-based consent management framework for healthcare domain is proposed in [28]. The study aims to achieve the GDPR's requirements. The experimental evaluations of the study are limited to the latency and throughput for read and write operations. The study does not provide any validation related to the audit mechanism. Besides, the study does not consider the data subjects' purposes.

A blockchain-based system for managing private data in a secure and decentralized manner is presented in [7]. The study highlights the privacy protection of user data from unauthorized access. The limitations of this study are regulatory compliance, interoperability, and integration with existing data management infrastructure. The authors present the prototype implementation and evaluation results for the healthcare case study. Besides, the study does not include a purpose tree.

A blockchain-based Service-Oriented Architecture (SOA) designed for consent management, access control, and auditing purposes is introduced in [29]. The authors propose a framework that combines blockchain technology and SOA to enhance transparency, traceability, and accountability. The paper presents a prototype implementation and evaluates its performance. The limitations of this study include interoperability, regulatory compliance, and integration with existing systems and infrastructure. Finally, a comprehensive literature survey on GDPR-compliant blockchain solutions is presented in [30]. The analysis presented in the study concludes that the GDPR-compliant blockchain solutions related to the data subjects' consent management and the data subjects' rights are the less explored studies in the literature. Furthermore, the survey presented in [31] focuses on the gap between the blockchain and GDPR, and concludes that providing the compatibility between blockchain and the GDPR is significant.

A comparison of the current literature with the proposed study is presented in Table 1. As seen in Table 1, the most common aspect of the related works for blockchain-based consent management is generally based on permissioned networks, except for [20] and [21]. These studies both offer a public blockchain solution for consent management. Besides, immutability in blockchain brings uncertainties related to the right to complete erasure from the system. The study presented in [24] differs from other studies by storing hash data on the blockchain. However, the study lacks of GDPR compliance and an audit mechanism. Furthermore, access control models force the entities to adapt to the access models of the presented related works. This enforcement prevents these studies from offering a general solution that can be applied to any field. For instance, if the access model of the study is constructed for healthcare, it cannot be applied to financial services without modification. Thus, another shortcoming in the presented related works is that the proposed solutions are standalone applications. Organizations must implement the necessary integration modules for these systems to work on their platform. Therefore, organizations must conduct a costly and time-consuming integration process for these solutions to function on their systems. The proposed study aims to serve as a middleware between the organizations and data subjects. Thus, the organizations do not need to change anything in their existing architectures. This architectural design makes the integration process seamless for organizations.

IV. SYSTEM ARCHITECTURE

In this study, a blockchain based GDPR-compliant auditable hybrid consent management architecture is proposed. The proposed architecture consists of two main components: off-chain and on-chain components. Off-chain components include the system implementation, Representational State Transfer (REST) Application Programming Interface (API), relational database, and the purpose tree rules. On-chain components include a blockchain ledger to store states of the consent data. The proposed system has three actors: the data controller, the data requester, and the data subject.

The data controller maintains the consent management system and performs the data integrity operations between

TABLE 1. Comparison of the proposed system with the current literature.

Ref.	GDPR	Dynamic Consent	Audit	Type of Network	Blockchain Consent	Purpose Tree	Implementation	Performance Evaluation
[19]	✓	✓	✗	Permissioned	Dataset Profiles	✗	✓	✓
[20]	✓	✓	✗	Public	Consents	✗	✓	✗
[21]	✓	✓	✓	Public	Health Data Consents	✗	✓	✓
[22]	✗	✓	✓	Permissioned	Consents	✗	✗	✗
[23]	✗	✗	✗	Permissioned	Patient Data Consents	✗	✓	✓
[24]	✓	✓	✗	Permissioned	Hash Data	✗	✓	✗
[25]	✓	✗	✗	Permissioned	Health Data Storage	✗	✓	✗
[26]	✗	✗	✗	Permissioned	Health Data	✗	✓	✓
[27]	✓	✓	✓	Permissioned	Consents	✗	✓	✗
[28]	✓	✗	✗	Permissioned	Patient Data Consents	✗	✓	✓
[7]	✓	✓	✓	Permissioned	Consents, Access Logs	✗	✓	✓
[29]	✓	✓	✓	Permissioned	Consents, Audit Events	✗	✓	✓
Our Approach	✓	✓	✓	Permissioned	Hashed Consents, Reads, Audits	✓	✓	✓

the off-chain and on-chain components when an action is performed in the system. The data requester can query for the consented data by providing the purpose of the query. The data subject can manage her consent in the consent management system using the purpose tree provided for the consent. These components also include additional elements to accomplish an auditable GDPR-compliant system. The general architecture of the proposed system is shown in Figure 1. From the data subject's perspective, the consent management system works as shown in Figure 2.

The overall framework assumes that each participant keeps their access keys safely. This assumption enables absolute accountability in actions taken in the system. Personal data and actions performed in the system are logged in the database. Three operations are crucial and recorded on the ledger: creating a new consent, performing an audit, and reading consents. Merkle root hashes of these actions are calculated and inserted into the ledger.

A. OFF-CHAIN CONCEPTS

Off-chain components handle data transfer. Purpose tree, auditing mechanisms, access models, and consent management are elements of this component:

1) PURPOSE TREE

Privacy policies are closely related to the purpose concept. Thus, purpose plays a vital role in the privacy-protecting access control models. GDPR clearly states that personal data cannot be processed for any purpose other than the stated purpose by the data subject. A hierarchical structure based on the principles of generalization and specialization of purposes simplifies the management of permissions [32].

In this study, the use of the purpose tree keeps the data subject's interaction with the system at a minimal level. Thus, data subjects can consent to multiple purposes by consenting to a single purpose. The purposes' levels are used to traverse purpose tree on lower levels. If a data requester requests for a level 2 purpose, the tree is traversed to include level 1 and level 0 purposed consents. Traversing process stops once the level 0 purpose is reached. This allows dynamic management of purposes without the need for authorization for each query. Each purpose defined in the tree is associated with data fields.

While leaves have less data fields associated to them, the level above them have all of them combined.

The purpose-based access to consented data is formulated as follows [32]:

Purpose is represented with P, the purpose for accessing data is represented with $AccessPurpose(AP)$ and the data that can be accessed for the given purpose is represented with $IntendedPurpose(IP)$.

Let $IP = \langle Allowed\ Intended\ Purpose\ (AIP)\ ,\ Prohibited\ Intended\ Purpose(PIP)\rangle$ denote an intended purpose, where $AIP \subseteq P$ is a set of allowed intended purposes and $PIP \subseteq P$ is a set of prohibited intended purposes.

$$AIP^{\downarrow} = \bigcup_{p_i \in AIP} Descendants(p_i)$$

$$PIP^{\updownarrow} = \bigcup_{p_j \in PIP} Ancestors\left(p_j\right) \cup \bigcup_{p_i \in PIP} Descendants(p_j)$$

When a subject seeks access to an object, the subject's AP is compared with the IP assigned to the relevant data record, and the access is granted.

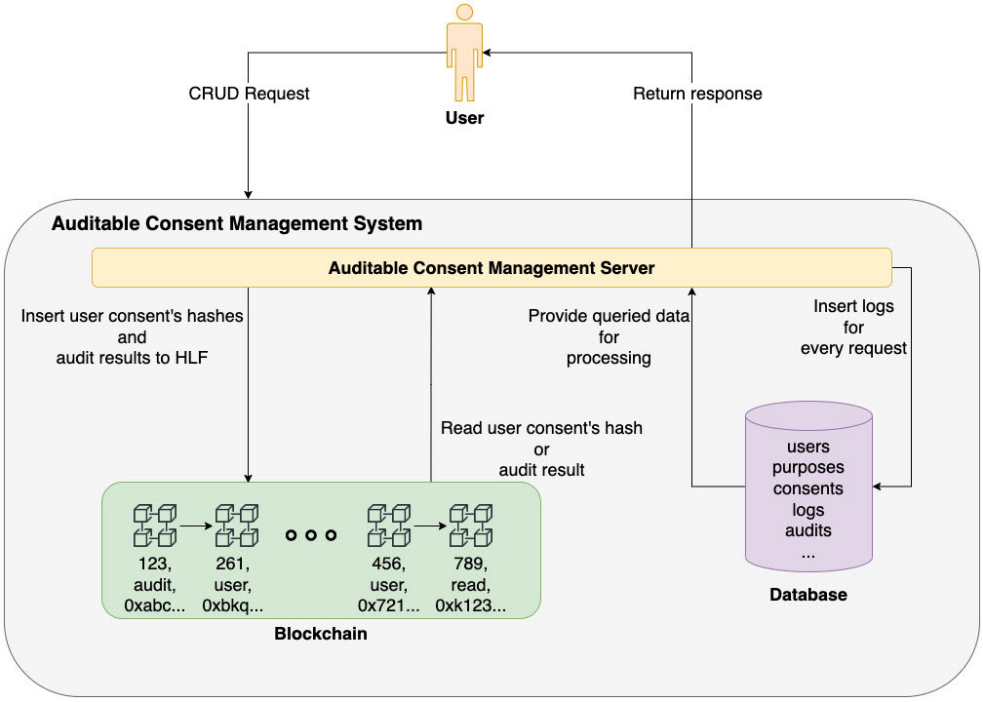

FIGURE 1. The general architecture and components of the proposed system.

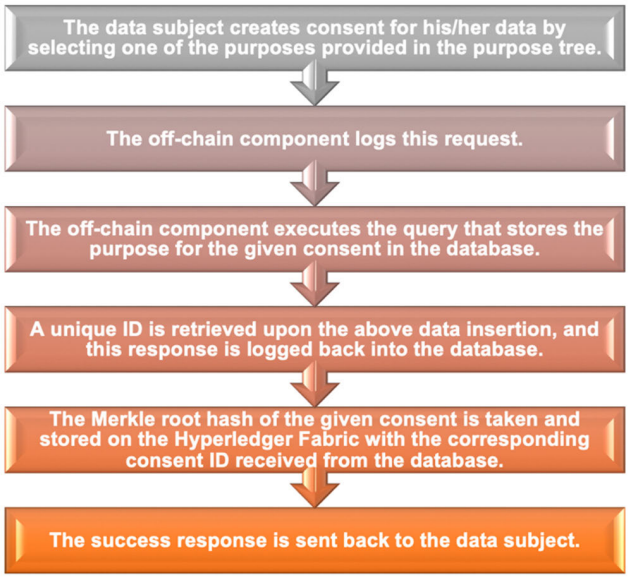

FIGURE 2. The workflow of consent management from the data subject's perspective.

2) CONSENT MANAGEMENT

In the proposed architecture, the user's consents are registered once, and necessary hashes are generated during creation time. These consents are stored on-chain and off-chain. Once the user creates a consent, it is recorded on the blockchain. This record represents the user's consent for future requests until it is expired or modified by the user. The recorded consent is reused in each query to expedite the query response time and save the system storage. In addition, if the user decides to change her consent, this is considered as a new consent. The newly created consent is linked to the user's previous consent in the database. This data is used in audits that are performed during that time interval. The newly created consent is inserted on the blockchain to be used for future queries.

The consent requests are in the form of Create, Read, Update, Delete (CRUD) requests and are handled by the server. Herein, the incoming requests are directly logged in the database. The calculations are performed for the response, and the calculated result is inserted into the ledger. Finally, the response is shared back with the user.

3) AUDIT SYSTEM

The audit system is the most essential part of the proposed architecture. The system needs on-chain and off-chain data to audit the previously performed queries. The audit system consists of two different types of audits: system audit and user audit. The system randomly triggers a system audit to ensure integrity and detect any violations that might have occurred in the system. User audit needs to be triggered by a user manually through the endpoint by sending a Hypertext Transfer Protocol (HTTP) request. If there is any violation in the system, then the user is immediately notified and provided with a list of violations in which the user's data is involved along with the violator's information. Audits can be performed in five audit modes:

- *Quick Audit:* Retrieves user's consent hashes from database and blockchain, and compares these two hashes with each other to verify the integrity of the consent. If these values are equal, the system assumes no violation has occurred and that both on-chain and off-chain components work in sync. This process can be performed instantly.
- *Full Audit:* All the user's consents are queried from the database, and the Merkle root hash of the consents is recalculated. This newly calculated hash is first compared with the consent hash in the database and then to the hash on the blockchain network. If all the hashes are equal to each other, the audit is inserted as a successful audit into the on-chain and off-chain components. If a violation is detected in this audit, the consent hash retrieved from the blockchain is accepted as the healthy consent hash, and the database is scanned for the last healthy consent. Once that consent is found, any other consents that are created after this are violation consents. The database is scanned for these violation consents to find out where these consents were created, and in which queries these consents were used to inform the user about the violations their data involved in. The system recovers the user's consent to the last healthy checkpoint, and the user's consent can still be used in the way it was supposed to be from the beginning after this audit and recovery process.
- *User Audit:* Allows the data subjects to perform audits on their access logs. The system tracks down every consent read performed on the data subject and checks the purpose of the consent read against the data subject's valid consent at the time of the consent read query. If any violation is detected in queries, the violators are tracked down, the violated consents are marked, and the authorities and the data subject are notified immediately with the derived information about the violation.
- *Log Audit:* It can be performed on any consent read log by a data subject in the related log or a data controller. Each consent in the access log is checked individually to ensure the purpose of the related query does not conflict with the data subject's valid consent as it was given.
- *Off-Chain Audit*: The data subject or the data controller can trigger off-chain audits. This audit is performed only on the off-chain side of the system. All the consents of the data subject are grouped, and the hashes registered for these consents are recalculated. The outputs are compared to the existing hashes recorded in the database. If any anomalies are found, the data subject is notified immediately.

4) DATABASE

Database stores various data in the system including authentication, authorization, personal information, consent preferences, purpose tree, purpose data fields, audit results, and logs. Privacy is established by storing sensitive information in a protected database. The flows in the consent management system are complicated since it works in sync with other components such as blockchain. Thus, the database must support transactions. It also needs to offer atomicity, consistency, isolation, and durability. In case of failed operations, the system must be able to revert to the initial state to maintain data integrity.

The consent read queries are complex and can include many parameters based on the query. Therefore, the database should scale well and perform optimized queries. Besides, consent modifications are not very often. Thus, consent query responses can be similar to each other. Therefore, the database requires a solid caching mechanism to perform well under the same queries, allowing faster read times.

B. ON-CHAIN CONCEPTS

There are two main parts of the on-chain component system: (i) the blockchain technology for storing user consents, consent reads, and audit outcomes, and (ii) the Merkle tree hashing algorithm to compress personal data into one hash to respect the user's *right to be forgotten*.

The blockchain technology can be used to develop the ideal solution that complies with GDPR and protects users from data violations. Figure 3 illustrates the blockchain architecture for the execution of a transaction. The smart contract side of the solution is implemented concerning GDPR rules. The system introduces an object called an *asset*. An asset can represent a user's consent, a query, and an audit entry. When a new user registers a consent, the smart contract requires the *userId* from the off-chain component, the Merkle tree root hash of the user's consent, and the asset type mentioned earlier. This entry is only meaningful when combined with the off-chain data since the system stores only an ID and a hash.

When a user requests to be removed from the system, this entry does not indicate anything on its own if the user is removed from the off-chain components. When a data read query is performed in the system, the system collects all the consents that match the query. These consents are grouped and Merkle root hash is calculated. The log ID received from the insertion of the data read is recorded on the ledger with the calculated Merkle root hash. When there is an audit entry insertion, the system requires the *auditId* from the off-chain side. Also, the Merkle tree root hash of the consent must be stored in the consent.

These components have update and getter methods that can be utilized when necessary, such as when the user wants to update her consent. Getter methods are primarily used in the auditing mechanism of the system when the records are being audited.

A Merkle tree is defined as a tree where each leaf represents a cryptographic hash of a given data block and they are hashed with each other until a single root hash represents all the data hashes from the beginning. This hashing method is frequently utilized on both off-chain and on-chain components. A user's consent representation is the Merkle tree root hash of that consent, which will be stored on the blockchain.

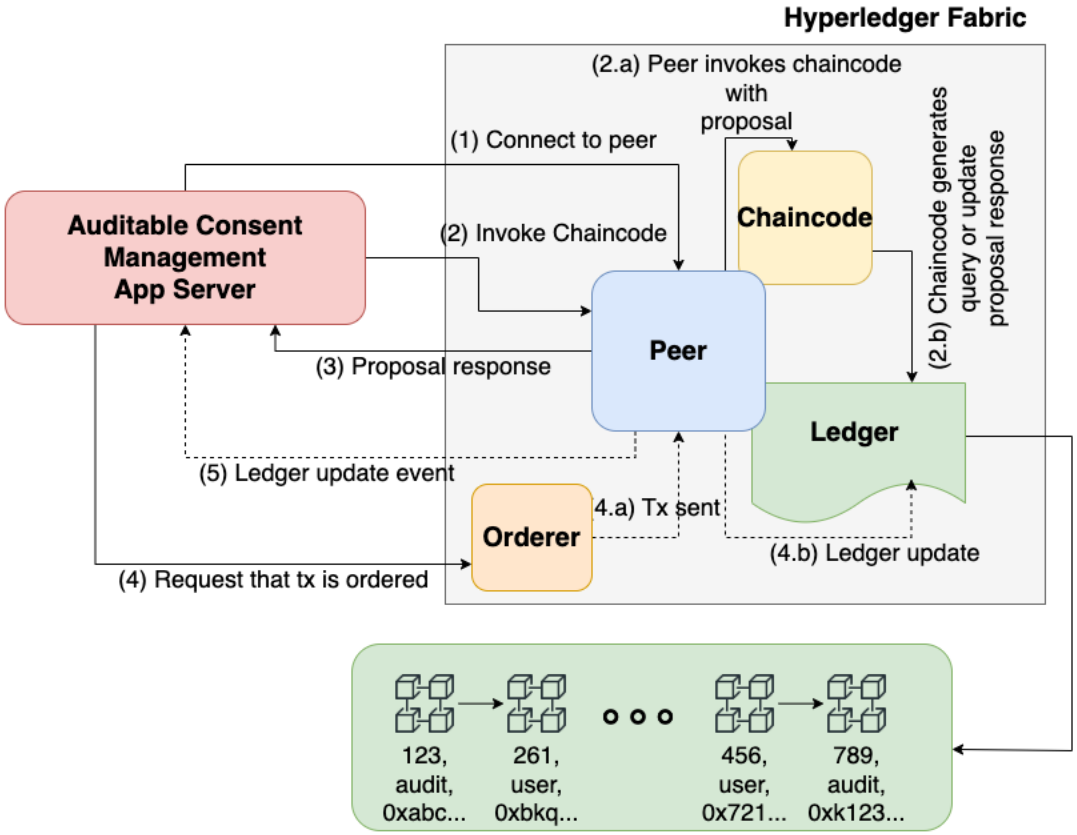

FIGURE 3. The blockchain architecture of the proposed auditable consent management.

V. EXAMPLE SCENARIOS

This section presents different violation scenarios and how the proposed system handles them. Also, it is assumed that blockchain access keys are stored safely since on-chain data is always accepted as the correct data to detect and recover violations.

A. INSIDER VIOLATION

An insider violation can mainly occur by a data controller in the system. A malicious data controller might change data subjects' consent. If the data subject's personal data is being shared for a purpose that is not defined by the data subject, then this is a clear violation of GDPR.

In this scenario, Bob is a data controller and Alice is a data subject. Alice has consented to purpose *Education*. Bob a malicious data controller takes advantage of his authority and changes Alice's purpose to *Business*. *Education* consent means sharing Alice's data such as *id*, *username*, *e-mail*, *birthplace*, *prefix*, *suffix*, and *maiden*, whereas *Business* means sharing data such as *id*, *e-mail*, *dob (date of birth)*, *gender*, *drivers*, *maiden*, *marital*, *race*, and *address*. This results in not only sharing Alice's personal information for a different purpose but also sharing personal information to which she never consented any purpose, such as *dob*, *gender*, *drivers*, *marital*, *race*, and *address*. This is a clear violation example of GDPR. An audit can detect this violation. In this context, there are several ways to trigger an audit in the system:

1. Random audits triggered by the system can occur at any time.
2. Alice triggers a user audit on her consent.
3. A user triggers a log audit on any consent read Alice's information is included in.

B. STOLEN CREDENTIALS

Stolen admin credentials may result in violations of consent preferences. Admins must keep their access keys safe. In such a case, a malicious user who possesses stolen admin credentials can perform changes to consent preferences. Therefore, an unauthorized party manipulates consent. Thus, this manipulation is a violation of GDPR.

In this scenario, Alice is an admin user in the system. Bob is a malicious user who stole Alice's access keys to the consent management system. Bob manipulates data subjects' consent preferences, resulting in GDPR violations. This kind of violation can only be detected by Alice. Bob has to use Alice's access keys to change the consents' purposes. As mentioned earlier, each operation in the system is logged by the system. These logs are kept in the database to be used in such cases. Once Alice realizes her keys are being used

for operations she has not performed, she can deactivate her keys by specifying the last operation she has performed in the system. The system collects each log where the *requesterId* corresponds to Alice starting from the given date of the last operation. All the operations performed by Alice starting from that date are marked as a *violation*. Therefore, each modification is restored to the last healthy state in the system. Hence, all the logs related to consent reads are collected and each data subject that exists in the logs is notified about the violation and its cause.

C. DATABASE VIOLATION

In database violation, a malicious user steals write access to database records and changes the data subject's personal information and consent preferences.

In this scenario, Bob is a malicious user who gained write access to the database. Alice is a data subject with the purpose of *Finance*. Bob first changes Alice's purpose from *Finance* to *Research*. Then, he changes Alice's *address* and *dob* information. Both of these operations are violations of GDPR.

The audit mechanism can detect this violation. Herein, all the audit types implemented in the system, except the quick audit, recalculate the user's consent hash. This calculated hash is compared with the stored hash on the blockchain. Hence, any direct change in the user information is detected when both hashes are compared. If there is a violation, every consent read performed on the manipulated consent is collected, and data requesters are also recorded to notify the data subject. The database write logs can then be examined to track down whose access keys are used to conduct this violation to remove access from the database. Afterward, if there exists any valid consent, the system attempts to recover the data subject's consent to the last valid consent.

VI. IMPLEMENTATION

In this study, Hyperledger Fabric [33] is used for the on-chain component of the proposed architecture. Hyperledger Fabric is an open-source project. It is a private and permissioned blockchain that can be used for use cases that require data privacy. Also, it is appropriate for protecting users from GDPR-related privacy violations. It comes with a unique consensus that the system admin can modify to fit the enterprise's needs. Therefore, Hyperledger Fabric is used to implement the on-chain component of the proposed architecture. The default configuration for the test network is maintained, except for block generation time which is set as ten milliseconds. Thus, a block will be generated by the blockchain system every ten milliseconds.

The database of the consent management system is built upon PostgreSQL. Also, concurrent connections are handled by Node.js. Further, the REST API is built using Express.js and the implemented components are served in Docker containers.

The purpose tree used in this study is shown in Figure 4. As seen, the purpose tree has three levels that are used to traverse the purpose tree. For query reads, the tree is traversed to obtain all allowed purposes in the consent query.

If a data requester requests for a level 2 *Academic* purpose, the tree is traversed to include purpose consents of *Research* and *All*. *Advertisement*, *Marketing*, and *Sales* are the same level purposes that share the same ancestor. While each purpose represents different data fields for consent, their ancestor *Business* represents all. The same flow is valid for the other leaves and their associations. The traversing process stops once the level 0 purpose is reached. The data fields obtained from the dataset are mapped to specific numbers as shown in Table 2. Table 3 presents the purposes that are matched with the related data fields.

When a data subject consents to *Finance*, it automatically gives consent for any purpose located below *Finance*. For example, if Alice consented to *Finance* when a data query for the purpose *DeFi (Decentralized Finance)* is issued, her data is also included in the query's response. If Alice's purpose was *DeFi*, then her data would not be shared in a query purposed for Investment, even if they share the same ancestor. This purpose tree allows data requesters to reach a wide range of user data and saves the data owners from having to consent for each purpose. The system makes all decisions based on this process.

Further, the Entity Relation Diagram (ERD) is constructed for the consent management flows. Figure 5 presents the related tables and their associations. The *users* table stores the user data and it is associated with the *consents* table. The *consents* table indicates information on the consented users. The *purposes* table enables the defined purpose to connect with its ancestors and descendants. Thus, the system traverses the tree while performing a query for the allowed purposes.

The *roles* table is used to store defined roles in the system. The *user_roles* table determines the user's role and establishes the relation between *users* and *roles*. The *data* table represents all data fields represented by *purposes*. A purpose can have multiple data fields that will be used to query for the consented purpose. The *data_purposes* table holds the information about purposes and establishes the relation between *purposes* and *data*.

The *audits* table stores the results of the performed audits in the system. Audits have predefined types as *system* and *user*. While system audits are randomly triggered by the system, user audits are performed upon a request from the user. The *status* field has predefined values such as *created*, *active*, *success*, and *violation*.

When an audit is first started, the status is *created* by default. The status is changed to *active* once the audit progresses in the system. If the result of the audit is successful, the status is changed to *success*. If it is unsuccessful, the status is marked as a *violation*. Thus, tracking of ongoing audits is achieved, and filtering of successful and unsuccessful audits in the system is enabled.

The *logs* table stores each step and action performed in the system. The *type* column of the table refers to the CRUD value of the log which is *create*, *read, update*, or *delete*.

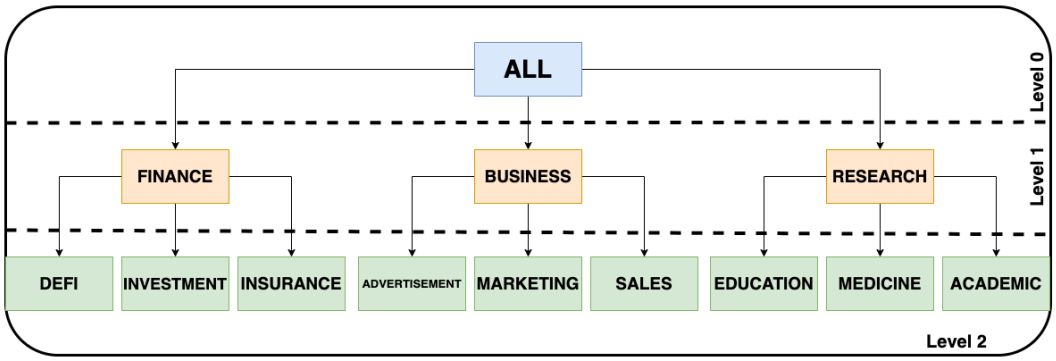

FIGURE 4. The purpose tree used for the implementation.

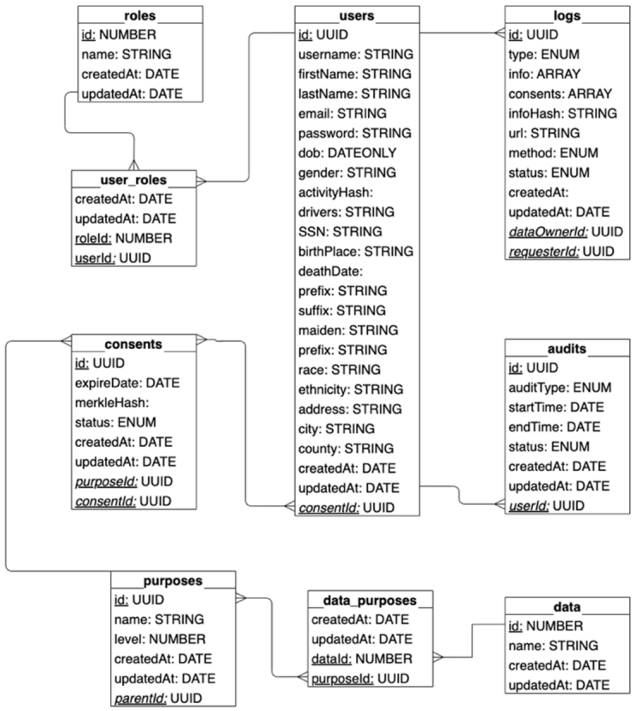

FIGURE 5. The ERD for the proposed consent management system.

The *method* column specifies the HTTP method of the log. For this purpose, the system supports GET, POST, PATCH, PUT, and DELETE methods. The *status* column indicates the health of the current log. If the log is found in a violation, its status is *violation*, otherwise it is *healthy*. *dataOwnerId* and *requesterId* refer to the user to whom the log belongs and who initiates the log, respectively. If the log is kept for a data read operation, then the *consents* field of the logs table is populated with every consent included in the response.

A database index is a data structure that prioritizes reads over writes and storage. Thus, lookup time can be optimized by using indexes. In the proposed system, the primary focus of the database architecture is to optimize queries performed in the system. Therefore, an index is defined on *expiryDate* of the *consents* table. As a result, the system performs faster query execution based on the *expiryDate*.

TABLE 2. The ID mappings for the data fields of the dataset.

Data Field	Mapping
id	1
username	2
firstName	3
lastName	4
email	5
dob	6
gender	7
drivers	8
SSN	9
birthPlace	10
deathDate	11
prefix	12
suffix	13
maiden	14
marital	15
race	16
ethnicity	17
city	18

A. ADDING A CONSENT

Figure 6 presents the flow for adding consent for the data subject. Each registered user is assigned a Bearer token during sign-up or when a valid login request is issued. A user can create consent by providing a purpose name and expiry date for her consent in a POST request. This allows the system to register the user's consent to be used in the data read requests.

The endpoint shown in Figure 7 presents an example HTTP request for adding consent.

As seen in line 4, a token is included in the header of the request to authenticate the user. In the body of the request, the data subject can specify the purpose and the expiry date for the given consent. If this is a valid request, the system returns a response. Otherwise, an unsuccessful response is displayed. This endpoint can also be used to update the consent that was registered earlier into the system. If an update is performed,

TABLE 3. The matching of purposes with the data fields.

Purposes	Data Fields																		
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*
Finance	X	X	X	X	X	X	X		X	X					X				X
Business	X				X	X	X	X						X	X	X			X
Research	X	X	X	X	X	X	X		X	X	X	X	X	X	X		X	X	X
DeFi	X	X				X	X												
Investment	X	X	X	X	X										X				X
Insurance	X	X	X	X	X	X			X	X					X				X
Advertise	X				X	X	X												
Marketing	X				X		X	X						X					
Sales	X				X		X								X	X			X
Education	X	X			X					X		X	X	X					
Medicine	X	X				X			X	X	X	X	X	X	X		X	X	X
Academic	X		X	X	X	X	X		X			X	X						X

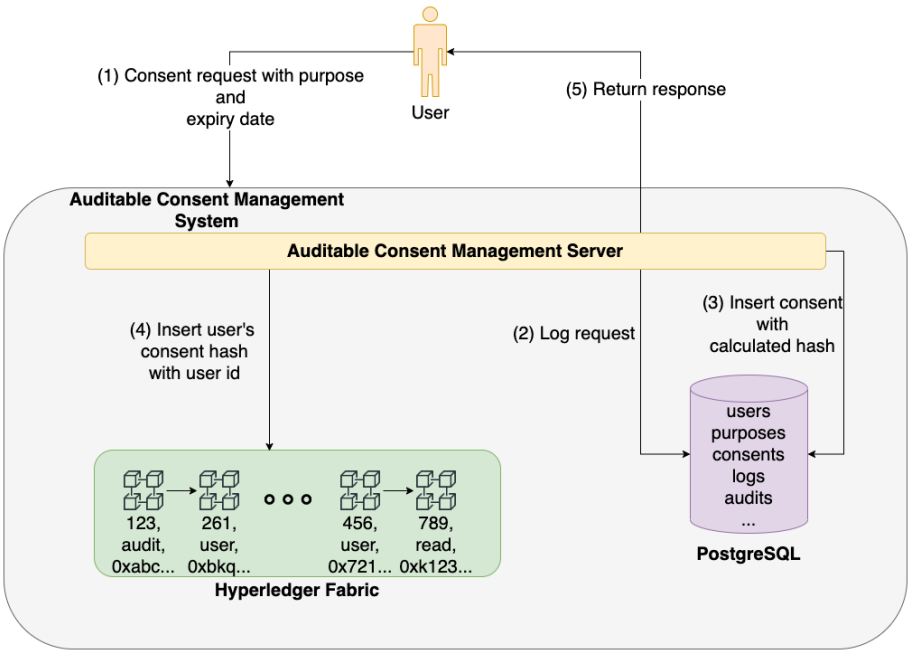

FIGURE 6. The system architecture for adding the data subject's consent.

the system marks the previous consent as expired. Thus, the system keeps track of all the changes.

The system logs the consent addition request and then executes a database insertion query after calculating the Merkle root hash. If insertion is successful, the consent hash and the corresponding *userId* received from the database are inserted on the ledger. Finally, the insertion result is returned to the user as a response.

B. GETTING CONSENTS

A user must be registered as a data requester in the system to be able to query user information. Figure 8 shows the system architecture for getting consented users. Each query must have a purpose in the request. The data subject also has the flexibility to query more parameters such as *expireDate*, *address*, *city*, *gender*, or other personal information. Thus, queries can be performed on specific user groups, such as users located in Los Angeles, or research on male data subjects.

Figure 9 shows an example Structured Query Language (SQL) query for a consent read. In the example shown, the data subject queries data subjects who have valid consents until 10.10.2050 for DeFi purposes.

The system processes this request with inclusion parameters. Data received from the database is processed, and the Merkle root hash of the consents is calculated to be inserted

```
HTTP
1 POST /api/v1/consent HTTP/1.1
2 Host: localhost:3000
3 Content-Type: application/json
4 Cookie:
      token=s%3AeyJhbGciOiJIUzI1NiIsInR5cCI6I
      kpXVCJ9.
      eyJuYW1lIjoidHVuYWhhbmRhZyIsInVzZXJJZCI
      6Ijc4MzMwZGQwLTBhODMtNGEzOC04NmQzLTYxZD
      UxODExNjNlYiIsInJvbGUiOlsiYWRtaW4iXSwia
      WF0IjoxNjg1Mjc5OTMyLCJleHAiOjE2ODUzNjYz
      MzJ9.
      jO_rcFxOZehU4Tq0jBXkTqLb9eU5S5zYQPvWtGl
      W_SY.
      C8RB1mEwBVhIzaE6x%2FTiHG3A%2B1UQRAjBLfa
      op7r6Kzo
5 Content-Length: 60
6
7 {
8     "purpose": "finance",
9     "expireDate": "10/20/2032"
10 }
```

FIGURE 7. The HTTP request for adding a consent.

into the Hyperledger Fabric. After the insertion completion, the query response is returned to the data requester. The consent read flow is represented in Algorithm 1. The time complexity for the Algorithm 1 is $O(n)$.

C. AUDITING

As mentioned earlier, audits can be performed in various audit modes. The full audit allows data owners to perform full audits on their data. The flowchart of the full audit is given in Figure 10. If the audit result is successful and no data violation is detected, a success message is returned to the user. Otherwise, the user's violated data, the queries in which this data was used, and the information of the violators are returned to the user with a response message. If a violation is detected, the user's consent is recovered to the latest healthy checkpoint in the system only if a valid consent exists. If there is no valid consent, it is set as empty. The time complexity of the full audit is linear.

The quick audit endpoint allows data owners to perform quick audits. The detailed flowchart of the quick audit is illustrated in Figure 11.

The flow of the quick audit is similar to the full audit. The main difference is that the quick audit only compares the final Merkle root hashes of the latest consent in the database and Hyperledger Fabric. If they are not equal, this is considered a violation of consent. If these values are equal, the system assumes no violation and that both on-chain and off-chain components work in sync. The quick audit process can be performed instantly. In the full audit, all the user's consents are queried from the database, and the Merkle root hash of the

Algorithm 1 Performing Consent Query

Initialization Parameters: *{query.params}*
Input: *A HTTP request with query parameters*
Output: *A HTTP response with consented users' data and count*

```
1  Function: readConsentedData():
2  purposeName ← query.params.purposeName
3  expireDate ← query.params.expireDate
4  purpose ← fetchPurpose(purposeName)
5  if(purpose == null) then
6      throw NotFoundError
7
   fields ← retrieveDataFields(purpose.id)
8  queries ← query.params.query
9  whereQueryForSequelize ← {}
10 for (index ← 0 to fields.length-1) do
11     element ← queries[fields[index]]
12     if(element != undefined) then
13         whereQueryForSequelize[fields[index]]← ele-
           ment
14
   ancestors ← []
15 ancestor ← purpose
16 while(ancestor.level != 0) do
17     ancestors.push(ancestor.name) // add the ancestor
         to array
18     ancestor ← findAncestor(parentId)
19
   ancestors.push("all")
20 purposes ← performQuery(whereQueryForSequelize)
21 if(purposes.length == 0) then
22     log ← saveLogToDatabase (url,req.method,[])
23
   else
24     log ← saveLogToDatabase(url,req.method,
       purposes.consents)
25
   hash ← merkleRootHash(purposes.consents)
26 submitTransaction(log.id,hash)
27 return purposes ← return the HTTP response with
     consented users' data
```

consents is recalculated. This newly calculated hash is first compared with the consent hash in the database and then to the hash on Hyperledger Fabric. If all the hashes are equal, the audit is inserted as a successful audit into the on-chain and off-chain components. Therefore, the full audit performs longer than the quick audit mode.

User audit allows data subjects to perform audits on their access logs. Figure 12 presents the flowchart of user audit. The system tracks each consent read performed on the data subject and checks the purpose of the consent read against the data subject's valid consent at the time of the consent read query. If any violation is detected in the queries, the violators are tracked, the violated consents are marked, and

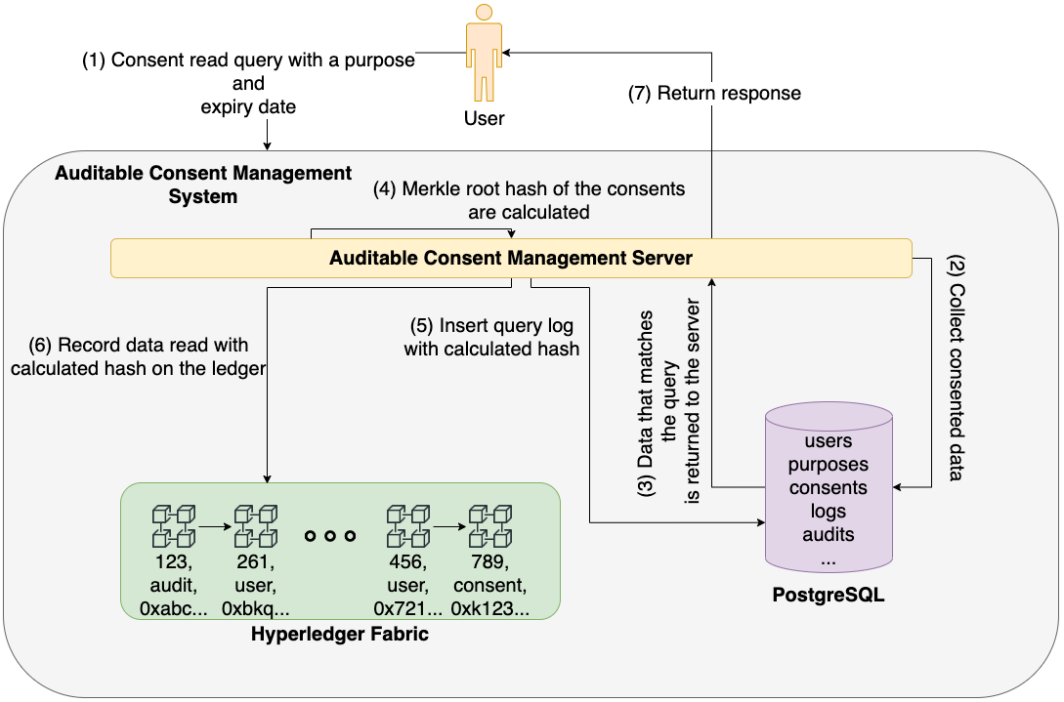

FIGURE 8. The system architecture for getting the consented users.

```
SELECT "purposes"."id", "purposes"."name", "purposes"."level",
"purposes"."createdAt", "purposes"."updatedAt",
"purposes"."parentId", "consents"."id" AS "consents.id",
"consents"."expireDate" AS "consents.expireDate",
"consents"."merkleHash" AS "consents.merkleHash",
"consents"."status" AS "consents.status",
"consents"."createdAt" AS "consents.createdAt",
"consents"."updatedAt" AS "consents.updatedAt",
"consents"."purposeId" AS "consents.purposeId",
"consents"."consentId" AS "consents.consentId",
"consents->users"."id" AS "consents.users.id",
"consents->users"."username" AS "consents.users.username",
"consents->users"."dob" AS "consents.users.dob",
"consents->users"."gender" AS "consents.users.gender" FROM
"purposes" AS "purposes" INNER JOIN "consents" AS "consents"
ON "purposes"."id" = "consents"."purposeId" AND
"consents"."expireDate" <= '2050-10-09 21:00:00.000 +00:00'
INNER JOIN "users" AS "consents->users" ON
"consents"."id" = "consents->users"."consentId" WHERE
("purposes"."name" = 'defi' OR "purposes"."name" = 'finance'
OR "purposes"."name" = 'all');
```

FIGURE 9. An example SQL query for getting a consent.

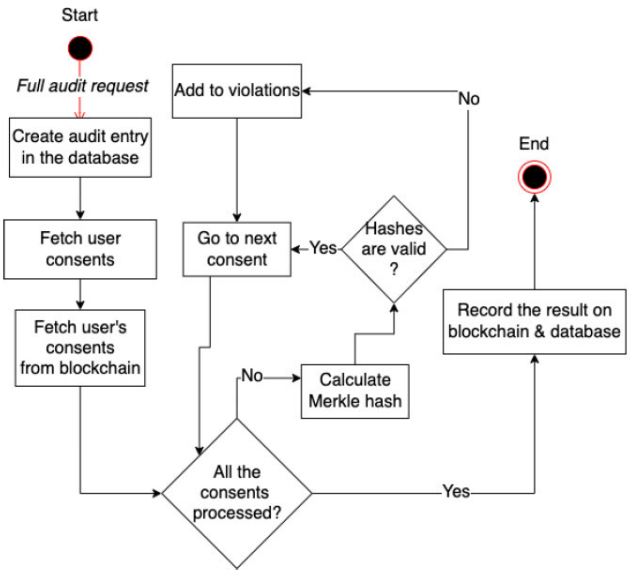

FIGURE 10. The flowchart of full audit.

the authorities and the data subject are immediately notified with the derived information about the violation.

The process of user audit is represented in Algorithm 2. The request for the user audit is triggered by the user or the system itself. Then, the system generates an audit entry with *date* and *auditType*. The user's consent is retrieved from the database along with all consent reads that the user's data is used.

The on-chain consent hash of the data subject is retrieved to be compared to the consent present in the database. As seen in line 6, both hashes are compared. If they do not match, this is detected as a violation. Later, the log gathered on the user is compared to purposes of consent queries with valid consent. If any of these do not match, that entry is appended to the violation list. Further, a last check is performed on the audit before the response. If any violations are detected, then

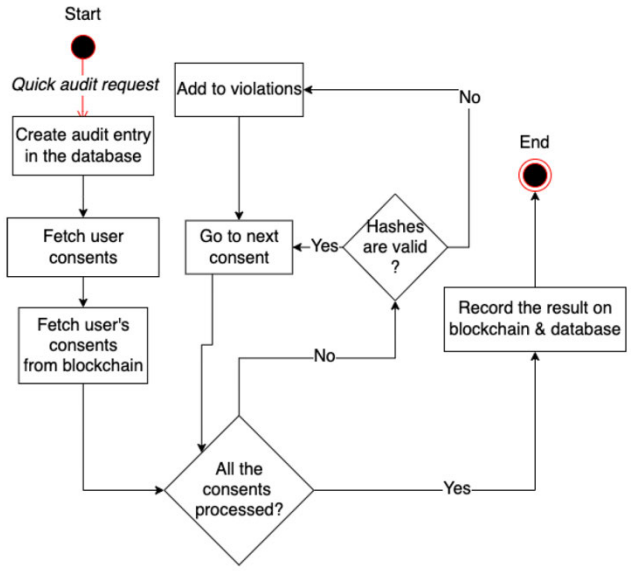

FIGURE 11. The flowchart of quick audit.

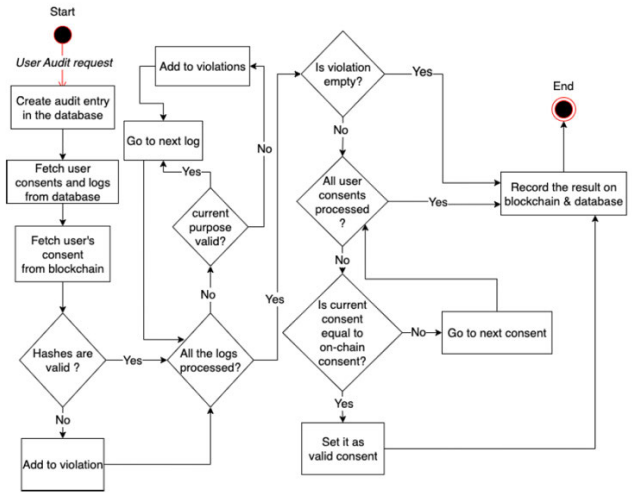

FIGURE 12. The flowchart of user audit.

the authorities and the data subject must be notified. Also, the data subject's consent preferences are recovered to the latest healthy consent to prevent further violations. Finally, the outcome of the audit is recorded on the ledger, and the response is shared with the user. The time complexity for the Algorithm 2 is $O(n)$.

The log audit can be performed on any consent read log by a data subject in that log or a data controller. Each consent in that access log is checked individually to ensure the purpose of that query does not contradict the data subject's valid consent at the time. The time complexity of log audit is $O(n)$. The flowchart of the log audit is illustrated in Figure 13.

Besides the mentioned audit modes, the data subject or the data controller can trigger off-chain audits. The off-chain audit is performed only on the off-chain side of the system. All the consents of the data subject are grouped, and the

Algorithm 2 Performing User Audit

Initialization Parameters: {auditType, date, userId}
Input: *A HTTP request with userId and authentication*
Output: *A HTTP response with audit results and information*

Function userAudit():
audit ← *createAudit(auditType,date,userId)*
userConsent ← *findConsent(userId)*
userLogs ← *fetchLogs(read,userConsent.id)*
onChainConsentHash ← *readConsentFromHLF(userConsent.id)*
violations ← *[]*
if(*onChainConsentHash != userConsent.hash*)**then**
  violations.push(userConsent.id)

fori ← *0 to userLogs.length-1* ***do***
  currentLog ← *userLogs[i]*
  purposeName ← *splitPurposeName(logUrl)*
  purpose ← *fetchPurposeFromDatabase(purposeName)*
  allowedPurposes ← *fetchAncestorsFromTree (purpose)*

if(*purpose == null) OR (purpose.id* ∉ *allowedPurposes*) ***then***
  violations.push(log.id,url,requesterId)

if(*violations.length >= 0*) ***then***
  userConsents ← *fetchPrevConsents(user)*
  status ← *false*
  fori ← *0 to userConsents.length-1* ***do***
    if(*usersConsents[i] == onChainConsentHash*)***then***
      userConsent ← *updateConsent(userConsents[i])*

submitTransaction(audit.id, status)
return *violations* ← *return the HTTP response with audit results and information*

hashes registered for these consents are recalculated. The outputs are compared to the existing hashes recorded in the database. If any anomalies are found, the data subject is immediately notified. Figure 14 presents the flowchart of the off-chain audit process.

Users are allowed to perform audits on all their consent reads. This audit includes not only one single consent read audit but also all of the consent reads for which the relevant data subject is present.

VII. RESULTS AND DISCUSSION

Experiments are conducted to evaluate the proposed consent management system. The scenarios given in Section V are

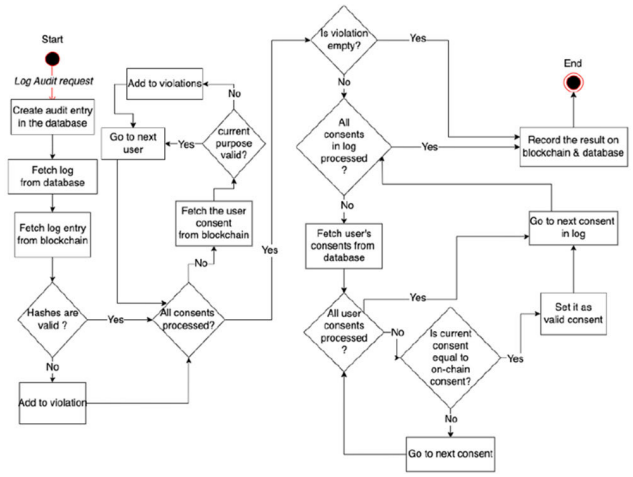

FIGURE 13. The flowchart of log audit.

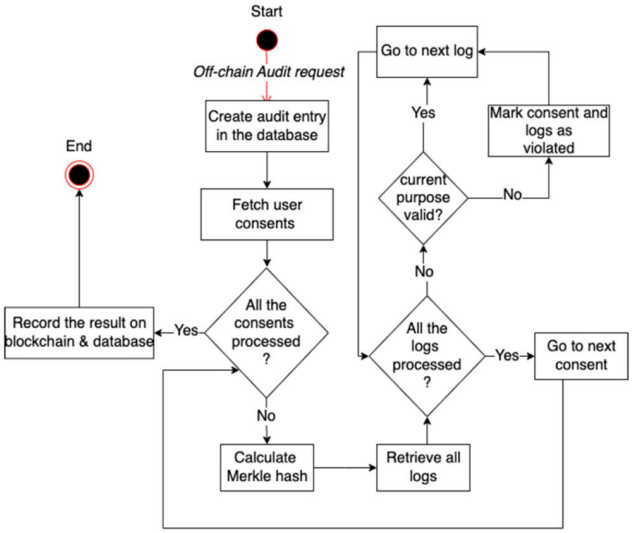

FIGURE 14. The flowchart of off-chain audit.

used for the experiments. Experiments are constructed to examine the scalability and throughput in terms of number of users and consents. Different consent settings are also applied, as well as the violations, to evaluate the audit mechanism and compare audit methods. Each experiment has run with a distinct amount of consent counts due to hardware limitations.

A. DATASET AND CONFIGURATION

SyntheticMass [34] data is used in the experiments. SyntheticMass is an open-source patient population simulation built by the MITRE Corporation [35]. For conducting the experiments, we parsed the dataset and formatted the data in a simpler form to feed into the system. A total of 10 million electronic records for synthetic patients are used in the experiments.

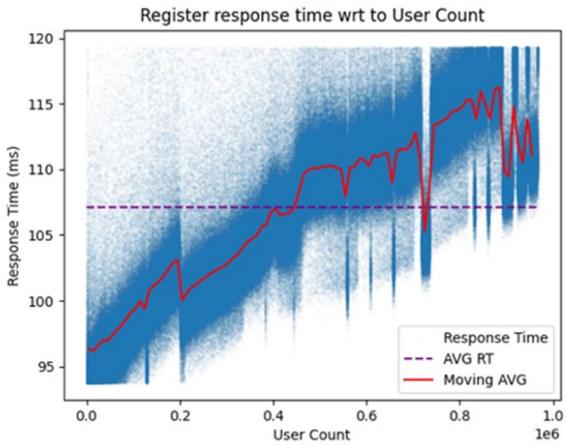

FIGURE 15. The register response time by the number of users.

The system is deployed in 11 Docker containers on a MacBook Pro 2019 machine with specifications of 2.6 GHz 6-Core Intel Core i7, Intel USD Graphics 630 1536 MB, and 16 GB 2667 MHz DDR4 RAM. The components in the docker containers are *cli*, *peer0.org1*, *peer0.org2*, *orderer*, *ca_org2*, *ca_org1*, *ca_orderer*, *dev-peer0.org1*, *dev-peer0.org2*, *posgres* and *consentMgmtSys*. The first nine containers are used for the Hyperledger Fabric network, the tenth container is the PostgreSQL database, and the last one is the consent management system implementation. Docker is allocated 6 CPU cores and 8 GB of memory.

B. USER OPERATIONS

The main user operations in the framework are register, login, and forget me. These tests were performed were performed using one million records. Figure 15 illustrates the registration of one million data subjects to the system. The request body includes personal information on the data subjects, such as their names, addresses, gender, and other personal information. Each request differs in terms of body size. We compared the response time as the number of users recorded in the system increased. The response times vary between 90-120 ms. The average response time across one million patient records is calculated as 107 ms. The moving average demonstrates that the system scales well, although the number of users has increased significantly, with the response time not increasing as much as the number of users. Therefore, it can be stated that the system handles the registration of millions of users. Also, the response time depends on the number of users registered at the time and the request's body size.

The login results seen in Figure 16 present a roughly constant line regardless of the number of users in the system. The response times vary between 75-79.5 ms. The difference between the response times is negligible for an HTTP protocol. Thus, this proves that the login part of the system scales perfectly independent of the number of users registered in the system. Therefore, even if we register a hundred million users, the average login time will be 77.5 ms on average. The

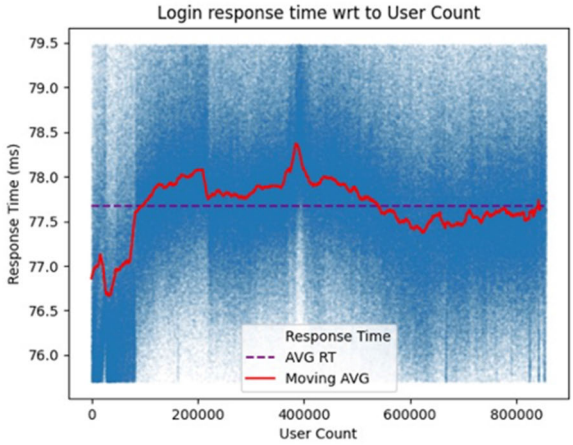

FIGURE 16. The login response time by the number of users.

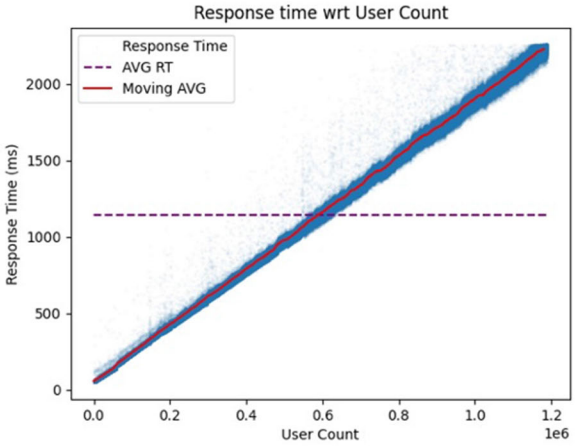

FIGURE 18. The response time of consent registration for users with random consent.

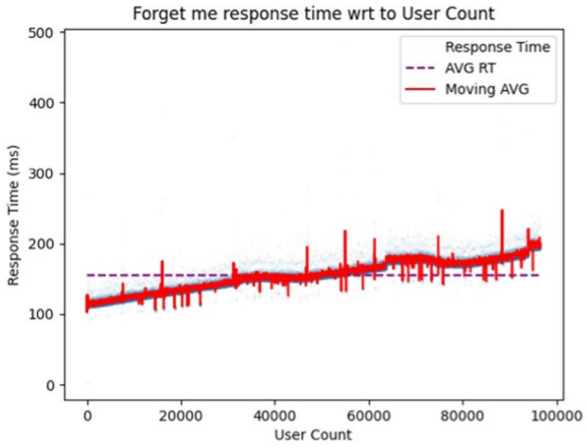

FIGURE 17. Forget me response time by the number of users.

irregular spikes may be caused by background activity within the system and can be discarded as the results vary within a range of approximately 3.5 ms. The response time depends on the body size of the request rather than the number of users currently registered.

Forget me endpoint is a critical component in the system to comply with the GDPR's *right to be forgotten*. 150.000 registered user records were deleted from the system to test this flow. Figure 17 shows that the results vary between 100 ms and 200 ms. The average deletion time of a user on 150.000 users is 150 ms. Thus, it can be concluded that as the patient record increases, the deletion operation time will also increase. Given the number of users, the response time is acceptable with the current hardware setup. Since this experiment mostly depends on the database configuration, the results can be improved with higher resource allocations.

C. CONSENT OPERATIONS

Regarding consent operations, 1.2 million consent registrations are displayed in Figure 18. These users have random purposes and random expiry dates for their consent. The average response time and moving average of the response time are also plotted in Figure 18. When the consent number reaches 1.2 million, the highest response time observed in the system is 2000 ms. The average response time of the consent addition is close to 1100 ms per user. The system's consent addition differs from the rest of the system endpoints. As seen in Figure 18, there is an increase in the response time as the number of consents increases. The reason for this is that the database indexes are implemented into the consent table on the expiration date of the consent. As a result, the system performs much faster consent reads since the reads are based on the consent's purposes and expiration dates.

Consent read queries is a crucial operation in the system. For this purpose, we have conducted three different experiments. Each experiment has consent from a different purpose level. For the related experiments, we have selected purposes as All, Finance, and DeFi. For this experiment, each request is run three times to calculate the standard deviation error for each consent count. Also, the experiments are based on 750 registered patient records and all the consents have the same expiry date. Figure 19, Figure 20 and Figure 21 show the results of these experiments. In Figure 19, all the patients have the purpose All for their consent. The purpose of All represents 19 data fields, which means every data field in the system.

Figure 20 displays the results for the purpose of *Finance* which is the level 1 of the purpose tree shown in Figure 4. *Finance* has 11 data fields associated with it. All the patients registered for this experiment have the *Finance* purpose for their consent. Finally, Figure 21 displays results for the purpose of *DeFi* which is under the *Finance* leaf and registered as a level 2 leaf in the purpose tree. *DeFi* holds 4 data fields.

In the results presented in Figure 19, 20 and 21, it is seen that the response time increases as the number of consents in the response increases. The main difference between these three experiments is the maximum and average response times. Hereby, it is concluded that if we have fewer data fields

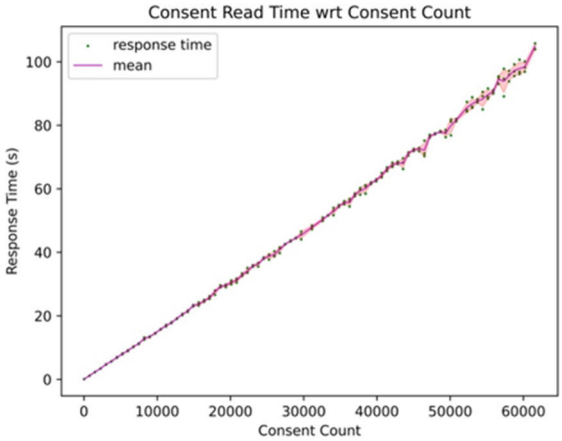

FIGURE 19. The consent read time for purpose level 0 (All).

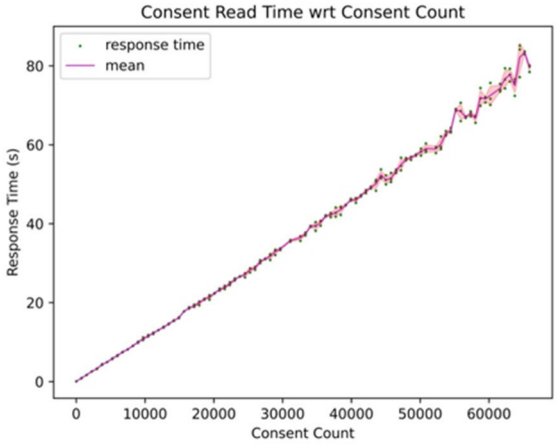

FIGURE 20. The consent read time for purpose level 1 (Finance).

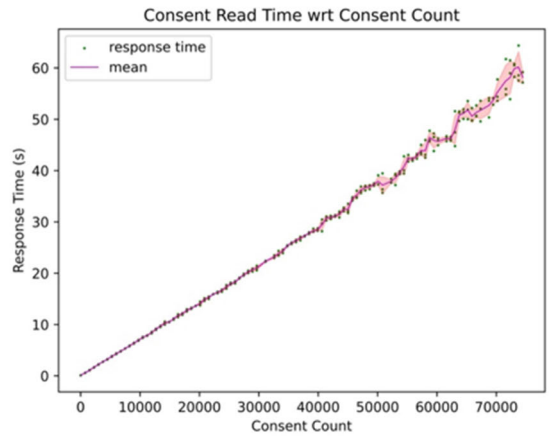

FIGURE 21. The consent read time for purpose level 2 (DeFi).

associated with the purpose, the response time is faster than the ascendant purposes' response times, even if they have the same consent count.

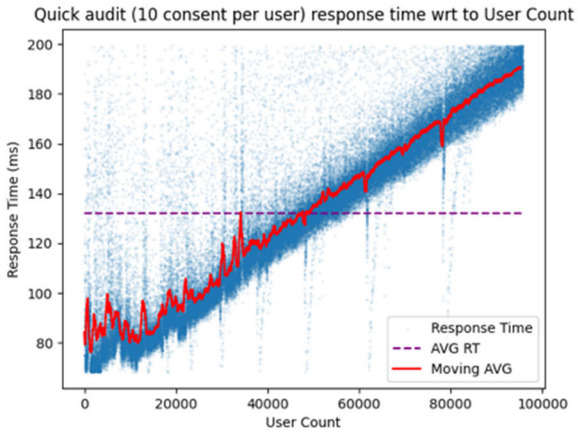

FIGURE 22. The response time for quick audit.

Table 4 presents the mean, median, maximum, minimum, and standard deviation of the related experiments. In the table, x columns represent the number of consents for each category, and y columns represent the operations' run times in seconds. The average response times are computed as 47, 38.07, and 27.39 seconds, respectively. Thus, it is observed that the system works significantly faster when handling fewer data fields regardless of the number of consents. Besides, as the purpose level goes down in the purpose tree, the response time will be faster, even if more records are being shared in the response. Furthermore, as the consent count increases, the response times get faster than in previous iterations. The reason for this is PostgreSQL's caching mechanism. If the same query is executed multiple times, the database retrieves the results from the cache enabling a faster response.

D. AUDIT OPERATIONS

Five different types of audits are implemented to audit data integrity and any possible violations that might have occurred with the user's consent.

Quick audit is tested with 10,000 patient records with each patient having 10 consent entries. In total, 100,000 consents are used for this experiment. The results of the experiment is given in Figure 22. As shown in the figure, the response time varies between 80 ms and 200 ms. The average response time computed is 132 ms. The spikes observed in the graph are from patients' different levels of purposes. A purpose with fewer data fields is audited faster than a purpose with more data fields.

The full audit is also conducted using 10,000 patient records with each patient having 10 consent entries. The obtained results are very similar to the quick audit as shown in the graph illustrated in Figure 23. The average response time computed is 132 ms for this experiment as well. Therefore, it is concluded that the time to recalculate the Merkle root hash is negligible. Furthermore, both algorithms have a linear time complexity. Consequently, the full audit offers a more

TABLE 4. Response times for reading consent.

	Mean		Median		Maximum		Minimum		Standard Deviation	
Level	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
0	30278.95	47.00	29660	45.14	61611	105.81	1	0.07	17911.61	29.49
1	32732.45	38.07	33320	37.45	65904	85.15	1	0.06	19345.03	23.53
2	36521.62	27.39	36987	26.69	74453	64.36	1	0.09	21603.04	17.07

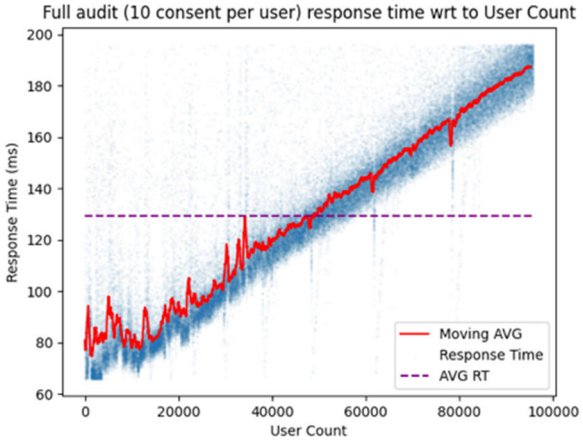

FIGURE 23. The response time for full audit.

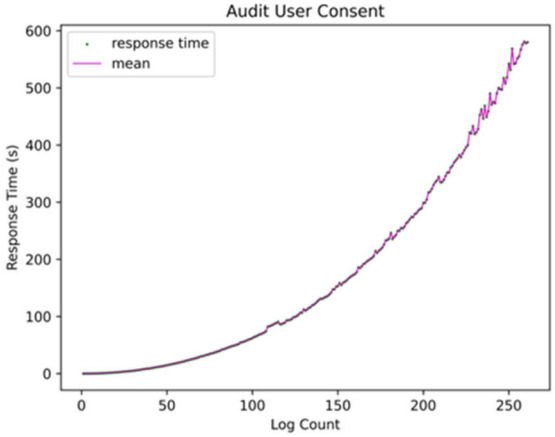

FIGURE 25. The response time of user audit for purpose level 1 (Finance).

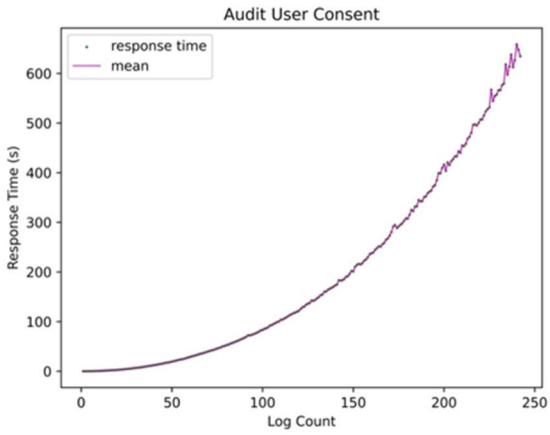

FIGURE 24. The response time of user audit for purpose level 0 (All).

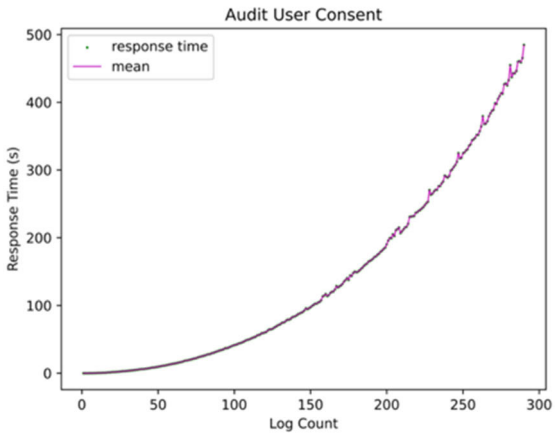

FIGURE 26. The response time of user audit for purpose level 2 (DeFi).

secure way of auditing consents by recalculating the hashes and comparing them to the on-chain data.

The user audit is tested with 100 patient records each patient has one consent registered in the system. Three different experiments are run for this audit type with different level purposes: *All*, *Finance*, and *DeFi*. The consent read queries are performed on the database and the user audit is called after each iteration. Hence, this increases the log count for each user. The obtained results for *All*, *Finance*, and *DeFi* are presented in Figure 24, Figure 25, and Figure 26, respectively.

The figures show that as the number of logged consents increases, the response time also increases. As seen in Table 5, the response time becomes faster as we descend the purpose tree. The average response times are computed as 192.27, 172.19, and 139.40 seconds, respectively. Therefore, it is concluded that for user audit, if the consent's purpose has fewer data fields associated with it, the audit response time will be faster compared to the purposes with more data fields.

The mean, median, maximum, minimum, and standard deviation of the related experiments are given in Table 5. In the table, *x* columns represent the number of logged

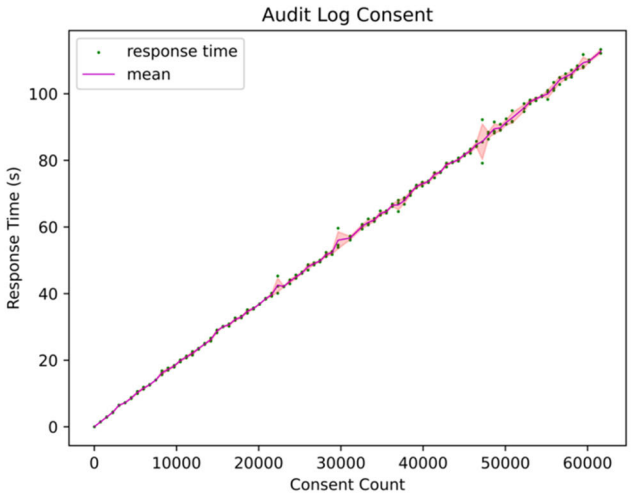

FIGURE 27. The response time of log audit for purpose level 0 (All).

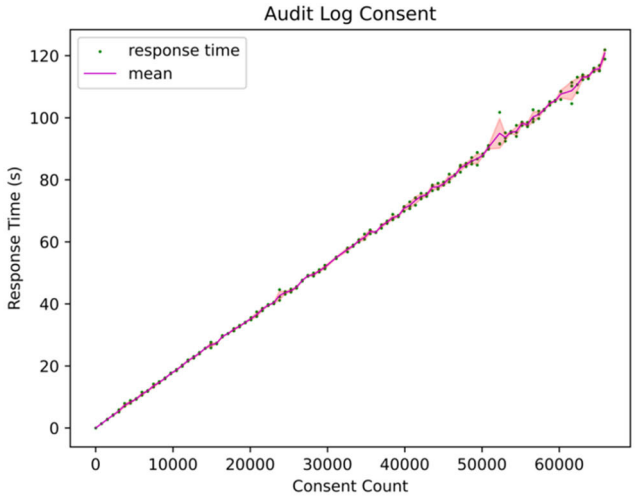

FIGURE 28. The response time of log audit for purpose level 1 (Finance).

consents, and *y* columns represent the run times in seconds. The overall results presented in Figure 25 are better than the results presented in Figure 24. This is an expected outcome since the level 1 purpose *Finance* has fewer associated data fields than the level 0 purpose *All*. Thus, the system fetches the consent data faster and performs the user audit on data faster than the earlier results. The results presented for level 2 are better than the results of level 0 and level 1. Hence, as the level increases in the purpose tree, the response time decreases since higher-level purposes have fewer associated data fields. In addition, all results show that the response time increases as the number of logged consent counts increases.

The log audit experiments are performed on 10,000 registered patient records. Three distinct scenarios are created with different purpose settings for patients' consent: All, *Finance*, and *DeFi*. For each 750-consent registration, three log audits are triggered to calculate the error margins and the response times. Figure 27 shows results for patients registered with the same level of purposed consent which is a level-0 purpose, *All*. The average response time is 55.29 seconds. It is observed in the figure that as the number of consents in a log increases, the response time also increases.

Figure 28 presents the results of the level-1 purpose, *Finance*. The average response time is 58.57 seconds which is a greater response time than the level-0 purposed experiment. This is because there are more consents in a log in this experiment. The pattern related to the increase in response time as the number of consents increases is also observed in this experiment. Finally, results of the last experiment with a level-2 purpose, *DeFi*, is shown in Figure 29. The average response time is 64.47 seconds.

Table 6 represents the results of these experiments. In the table, *x* columns represent the number of consents in the log, and *y* columns represent the run times in seconds. The results demonstrate that the purpose has no impact on the response time. The purpose and the level of purpose do not affect the

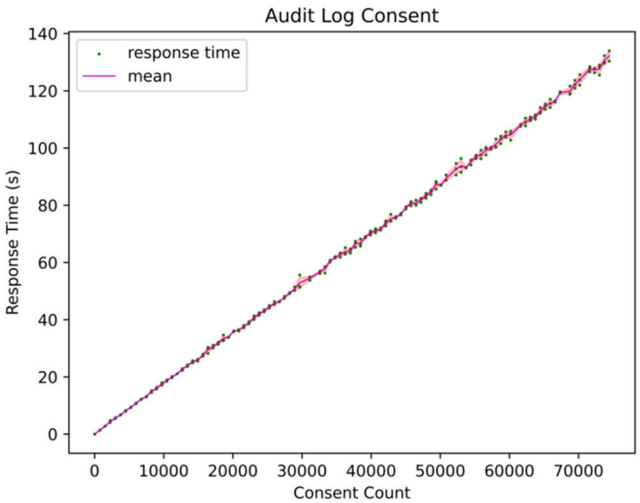

FIGURE 29. The response time of log audit for purpose level 2 (DeFi).

response time because log audit checks the valid consent at the time of the consent read's purpose.

In the previous experiments, it was observed that the purpose level directly affected the response times as they had more data fields associated with it. For the log audit, the number of consents in the log is the primary parameter. Because the audit log retrieves the consent and verifies that the purpose of the consent and the purpose of the query match. Therefore, the purpose level does not affect the response time. The data fields associated with the purposes are not crucial in the log audit. Hence, the level of purpose is not a determining parameter.

The last experimental setup is initiated for the off-chain audit. These experiments are performed with 10,000 patient records. After each consent registration, an off-chain audit is triggered, and three different level purposes are added individually to each experiment to observe the behavior of different purpose levels. Hence, a purpose from each level is

TABLE 5. Response times for user audit.

	Mean		Median		Maximum		Minimum		Standard Deviation	
Level	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
0	121.5	192.27	121.5	127.19	242	685.56	1	0.014	69.86	187.28
1	131	172.19	131	112.26	261	580.98	1	0.012	75.34	169.28
2	145.5	139.40	145.5	91.61	290	484.73	1	0.018	83.72	135.96

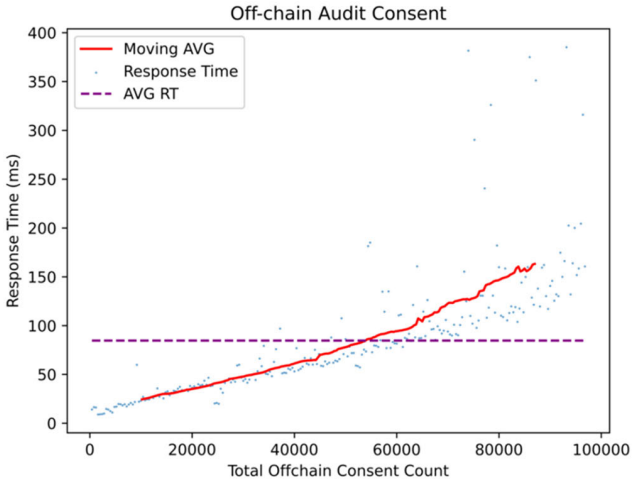

FIGURE 30. The response time of off-chain audit for purpose level 0 (All).

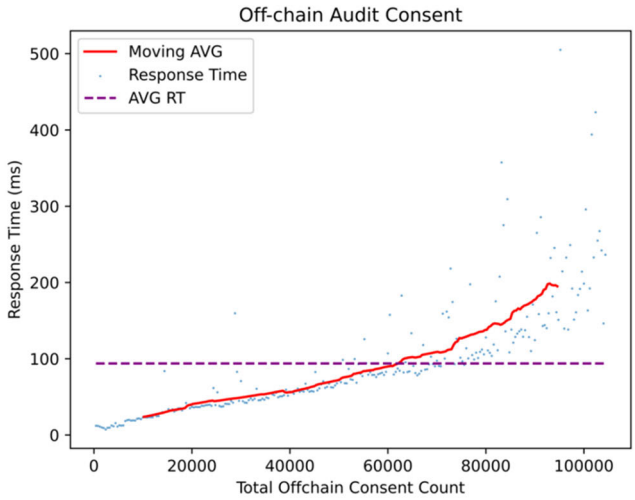

FIGURE 31. The response time of off-chain audit for purpose level 1 (Finance).

selected for the off-chain audit experiments: *All*, *Finance*, and *DeFi*.

Figure 30 presents the results of the level-0 purpose, *All*. The spikes in the results are negligible due to the network distribution and the background activity in the hardware.

Figure 31 presents the results of the level-1 purpose, *Finance*. The obtained results are similar to Figure 30. In this experiment, contrary to Figure 30, the response times increase. The reason for this is the increase in the number of consents.

The results presented in Figure 32 are the level-2 purpose, *DeFi*. This graph also has a similar pattern to Figure 30 and Figure 31.

Table 7 represents the results of the off-chain audit experiments. In the table, *x* columns represent the number of consents, and *y* columns represent the run times in seconds. According to the results of these experiments, the purpose level is not a direct parameter in the off-chain audit type. Regardless of the level of the purpose, the outcomes are in the same pattern. Besides, it is concluded that the purpose and the associated data fields are not directly related to the off-chain audit, but to the consent object itself. Therefore, the obtained results are consistent with the expectations.

Further, an ablation test is performed to present the impact of the audit with and without blockchain. Therefore, the ablation study was conducted to compare the audit process with and without the blockchain components. For this purpose,

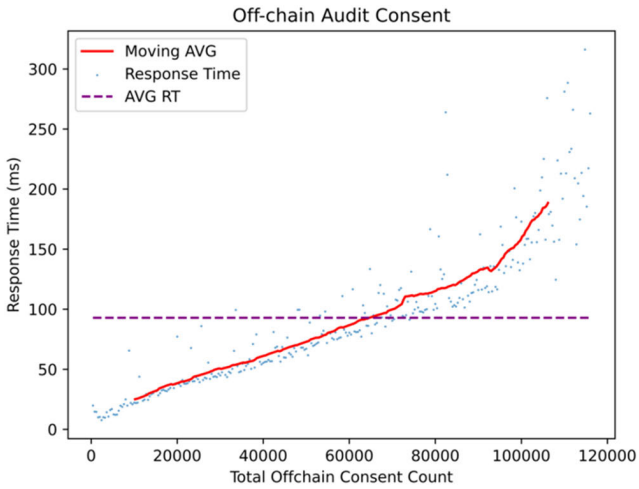

FIGURE 32. The response time of off-chain audit for purpose level 2 (DeFi).

the user audit is selected for this analysis and a dataset of 100 users with 70 consents per user is utilized. The results seen in Figure 33 show that using blockchain components maintains the constant execution time. This constancy is attributed to Hyperledger Fabric's architecture, where assets are represented as key-value pairs, and efficient key-based lookups are enabled for querying and updating the ledger.

TABLE 6. Response times for log audit.

	Mean		Median		Maximum		Minimum		Standard Deviation	
Level	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
0	30,278.95	55.29	29,660	54.21	61,611	113.30	1	0.017	17,911.61	32.75
1	32,732.46	57.45	33,320	58.57	65,904	116.88	1	0.015	19,346.02	33.64
2	36,521.62	63.82	36,987	64.47	74,453	133.94	1	0.018	65,975.6	115.43

TABLE 7. Response times for off-chain audit.

	Mean		Median		Maximum		Minimum		Standard Deviation	
Level	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*	*x*	*y*
0	48,600	84.61	48,600	70.85	96,800	385.20	1	0.018	27,943.51	64.54
1	52,400	93.92	52,400	78.74	104,400	505.01	1	0.02	30,317.46	75.01
2	58,200	92.92	58,200	81.32	116,000	316.20	1	0.019	33,486.12	59.41

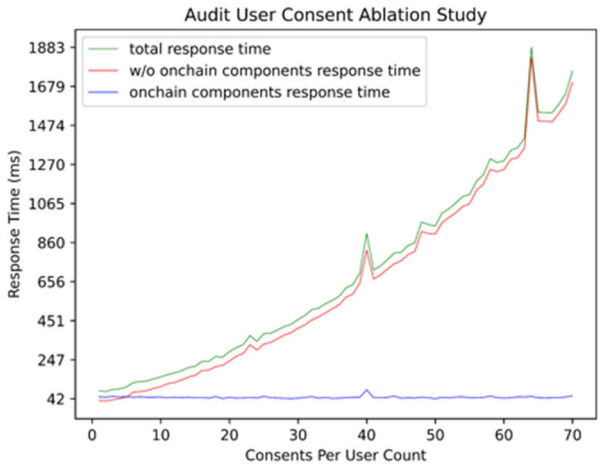

FIGURE 33. The impact of the auditing consents with and without blockchain technology.

The average response time measured in the ablation study is 49.26 ms. As the number of consents in the system increased, the response time for on-chain components remained constant. Thus, this demonstrates the efficiency of the system's design, specifically, the use of purpose trees and the strategy of storing only hashes on the blockchain. These optimizations significantly reduce the computational burden commonly associated with blockchain technologies, ensuring the proposed solution remains effective and resource-efficient.

E. HARDWARE RESULTS

This section analyses the hardware performance of the proposed system. Two performance graphs are drawn while performing the consent addition and reading consented data from the system. Figure 34 shows that the CPU core utilization is 51.26%, with 400% CPU allocated. 7.68 gigabytes (GB) of RAM is allocated for these experiments, where only 631.74 megabytes (MB) is used for registering consents at the consent registration time. It is seen that the disk read/write is close to 5.5GB, which is acceptable in terms of consent registration. The figure presents the time that a large chunk of data is inserted into the system. Hence, Figure 34 shows that the system scales well at 1.2 million consents.

At the time of the query, there is an increase in CPU usage since the system performs an I/O operation. Besides, out of 400% allocated CPU, only 110% has been used. After performing the query, the CPU usage was 51%. This is the system in idle status. Even if no operations are being executed at the time, 11 containers contain the database and the related Hyperledger Fabric nodes. Memory usage is also only at 631.74 MB, whereas 7.68 GB RAM is allocated for the system. The system manages consent registration exceptionally well, even at 1.2 million consents.

Figure 35 presents the hardware usage of a consent read. It can be observed that the CPU usage is dropping down to 55% from 110%, and the memory usage is increasing to 1.2 GB from an idle position of 0.9 GB. As seen in Figure 34, reading consent consumes more resources.

These performance evaluations are given for an estimation of minimum system requirements. The results prove that a machine with 4 CPUs and 8 GB RAM can manage up to 1 million consents.

VIII. FINDINGS AND IMPACTS

This study confirms that blockchain integration achieves a constant response time. In addition, it is important to acknowledge the broader advantages of blockchain technology that are outlined below:

- *Immutability:* Once recorded, data on the blockchain cannot be altered or deleted. This immutability ensures the integrity and trustworthiness of the consent records.

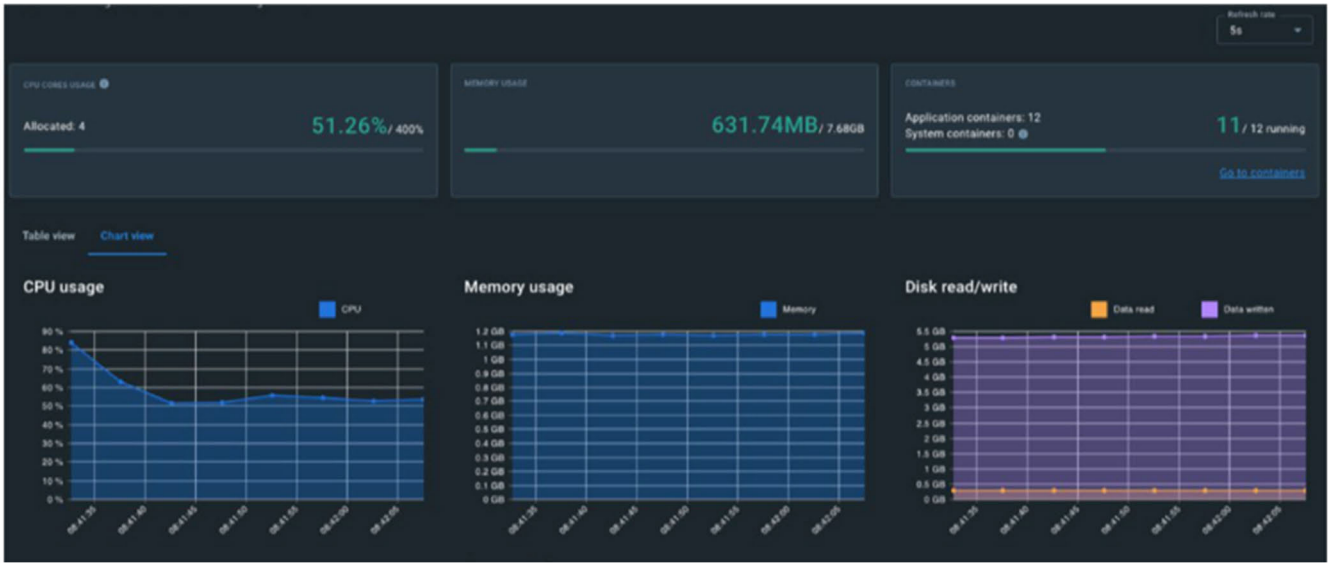

FIGURE 34. The hardware performance for registering consents.

- *Transparency*: Blockchain provides a transparent ledger that can be audited by all stakeholders, enhancing trust and accountability in the system.
- *Decentralization*: The decentralized nature of blockchain removes the need for a central authority, reducing the risk of single points of failure and enhancing system resilience.
- *Security*: Blockchain's cryptographic principles provide a high level of security, protecting data from unauthorized access and cyber threats.
- *Traceability*: Every transaction on the blockchain is traceable, allowing for complete visibility into the history of user consents and any modifications.
- *Efficiency in Compliance*: Blockchain simplifies regulatory compliance by providing a clear and auditable trail of users' consents, reducing the administrative burden for organizations.

Hence, considering the benefits of the blockchain technology, it is not advantageous to forfeit this technology to achieve constant response time. The implementation of blockchain and the optimizations demonstrated in this study offer an effective solution that ensures data integrity, transparency, and security without compromising performance. Thus, integrating blockchain technology into the audit process provides a robust and comprehensive approach to managing users' consent effectively.

In this study, blockchain technology is integrated into the audit phase of user consent management, providing a robust framework for ensuring data integrity and transparency. The implications of this work are multifaceted:

- *Enhanced Data Integrity and Security*: The proposed solution leverages blockchain to ensure that all approval transactions are recorded in an immutable and secure manner. Therefore, the risk of unauthorized modifications is eliminated and trust in the audit process is enhanced.
- *Transparency and Trust*: The decentralized nature of blockchain provides an open ledger that can be audited by multiple stakeholders. This transparency increases trust among users, organizations, and regulators as consent records are tamper-proof and verifiable.
- *Efficiency in Compliance*: The proposed system simplifies the process of regulatory compliance by providing a clear and auditable trail of user consent. Therefore, the administrative burden on organizations is reduced and adherence to data protection regulations like GDPR and California Consumer Privacy Act (CCPA) is ensured.
- *Scalability and Performance*: The proposed study demonstrates that the integration of blockchain does not adversely affect the system's performance. The proposed solution can scale effectively while maintaining constant response times even as the number of permissions increases, with the optimizations such as purpose tree and storing only hashes on-chain.
- *Cost-Effectiveness*: Although blockchain solutions are often criticized for their resource intensity, the optimizations implemented in this study mitigate these concerns. The use of Hyperledger Fabric's key-value pair system and efficient querying mechanisms ensures that the computational overhead is minimal.

As a result, this study provides several valuable lessons for the implementation of blockchain in audit processes:

- *Optimization*: The study highlights the importance of optimizing blockchain interactions. The proposed solution provides the balance between security and performance by using a purpose tree and storing only hashes on the blockchain. Thus, it is shown that with careful

FIGURE 35. The hardware performance for reading consents.

design, the resource requirements of the blockchain can be managed effectively.

- *The Blockchain's Role in Auditing*: The ablation study underscores the significant impact of blockchain in the auditing process. Hence, the study shows that the blockchain can provide constant execution time. Thereby, the audit process becomes more reliable.
- *Resource Management*: This study shows that despite the blockchain's reputation for high resource consumption, a resource-efficient solution can be implemented with optimization. This is essential for real-world applications where resource limitations are frequently an issue.
- *User-Centric Design*: The effectiveness of the proposed solution in managing and controlling user consent demonstrates the importance of designing systems that prioritize user transparency and control. Hence, the user-centric approach not only enhances compliance but also enhances user trust and engagement.
- *Scalability*: The findings indicate that the proposed solution can handle increasing numbers of consents without degradation in performance. This scalability is essential for real-world applications where the data volume increases rapidly.

IX. CONCLUSION AND FUTURE WORK

In this study, a novel approach is proposed to existing consent management systems. The proposed GDPR-compliant hybrid architecture has the strengths of both blockchain and database technologies and integrates the purpose tree into the consent management. Thus, the proposed study achieves decentralization, immutability, and auditability to detect and correct possible violations.

The following contributions are achieved with this study:

i. *Implementing a GDPR-compliant consent management system using blockchain and purpose tree*: A private blockchain, Hyperledger Fabric, is used as the blockchain technology. A purpose tree implementation is introduced and integrated into the consent management system to manage data subjects' purposes. The purpose tree enables the data subjects and the system to manage consents dynamically and offers a seamless user experience to the data subjects in terms of managing their consents.

ii. *Utilizing both on-chain and off-chain technologies*: The blockchain provides immutability, accountability, tamper-proof record keeping, and preserves privacy to be GDPR-compliant. Thus, the audit results, consents, and consent queries are hashed and stored in Hyperledger Fabric. On the other hand, using a relational database allows the system to perform complex queries that do not scale well with the blockchain architecture. Personal data collected from data subjects are stored in the database to allow them to be erased entirely from the system at any time they require. This cannot be achieved by solely using blockchain technology due to its characteristics. The only remaining data from data subjects' personal information is the hash which has no meaning by itself. Thus, GDPR's *right to be*

forgotten is achieved. Therefore, the shortcomings of both technologies are replaced with each other to make the consent management system GDPR-compliant.

iii. *Implementing auditability to detect violations*: This objective is achieved by implementing five different auditing mechanisms. These audit types are listed as quick audit, full audit, user audit, log audit, and off-chain audit. For these audit mechanisms, experiments were run and the obtained results were evaluated.

iv. *Improving performance*: The time and cost of blockchain for auditing GDPR-compliant data are optimized. As a result, the proposed study presents an efficient blockchain-based solution for consent management and it can be applied to real-world applications.

The proposed system is designed as a middleware to reduce organizations' integration costs to make their systems GDPR-compliant. Therefore, organizations do not have to implement components on their existing system to integrate with the proposed solution. The system can be easily integrated into the existing systems by positioning it as a middleware between the existing system and users. Hence, this study ensures GDPR compliance by allowing the following eight fundamental rights:

- *Right to be informed (GDPR Articles 12 to 14)*: The system is designed to notify the data subjects immediately when a violation is detected in their consent. They can also query for their consent, personal data, and how and where they were used.
- *Right to access (GDPR Article 15)*: Any data subject is allowed to access any data related to them, including the logs stored in the system, how many times their data was shared, and for what purpose it was processed.
- *Right to rectification (GDPR Article 16)*: The data subjects have the right to rectify the data if data is found inaccurate.
- *Right to be forgotten (GDPR Article 17)*: The forget me functionality in the system enables data subjects to remove themselves from the system. The remaining hashes do not have a meaning without their personal data. Thus, there is no trace of their personal data in the system.
- *Right for data portability (GDPR Article 20)*: The system is designed to maximize the data subject's user experience. The data subject only enters or shares her personal data once with the system, and it is stored in a portable way where the data subject can reuse this data for different purposes across different services.
- *Right to restrict processing (GDPR Article 18)*: This right is achieved by using the purpose tree. The data subjects can configure their purposes for their consent to restrict the processing of their personal data.
- *Right to withdraw consent (GDPR Article 7)*: The data subject can set the consent invalid or expired at any time. If the data subject would like to delete her personal data from the system, she can also benefit from the right to be forgotten.
- *Right to object (GDPR Article 21)*: The data subjects can object to the use of personal data by following up on an audit's outcome or at any time they require.
- *Right to object to automated processing (GDPR Article 22)*: The data subjects can also object to automated processing by determining their purposes accordingly. This prevents the organizations from profiling the data subject.

In this study, the purpose tree is managed by the admin users. As a future work, dynamic purpose tree implementation will be developed to allow data subjects to create their purposes in the system. Also, the experiments were run on a single computer. Thus, the same experiments could also be run on different hardware with different configurations. In this study, there was only one setup of Hyperledger Fabric for the shared results due to hardware limitations. This could also be improved by deploying multiple nodes across different devices to generalize solutions in a real-world scenario where entities would possess individual nodes.

In conclusion, this study presents a novel hybrid GDPR-compliant and accountable dynamic consent management approach by using a purpose tree and blockchain technology. Therefore, the proposed approach provides an end-to-end GDPR-compliant experience to all parties involved in a consent management system. Besides, this study aims to be a blueprint and a guideline for future implementations in this field.

REFERENCES

[1] (2016). *EU's General Data Protection Regulation (GDPR). 2016/679 Of the European Parliament and of The Council of April 2016*. Accessed: Jul. 10, 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[2] IBM Security. (2022). *Cost of a Data Breach Report 2022*. Accessed: Jul. 10, 2024. [Online]. Available: https://www.ibm.com/downloads/cas/3R8N1DZJ

[3] R. R. Agarwal, D. Kumar, L. Golab, and S. Keshav, "Consentio: Managing consent to data access using permissioned blockchains," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Toronto, ON, Canada, May 2020, pp. 1–9, doi: 10.1109/ICBC48266.2020.9169432.

[4] E. Olca and O. Can, "DICON: A domain-independent consent management for personal data protection," *IEEE Access*, vol. 10, pp. 95479–95497, 2022, doi: 10.1109/ACCESS.2022.3204970.

[5] General Data Protection Regulation (GDPR). *Consent*. Accessed: Jul. 10, 2024. [Online]. Available: https://gdpr-info.eu/issues/consent

[6] IBM. *Overview of Consent Management*. Accessed: Jul. 10, 2024. [Online]. Available: https://www.ibm.com/docs/en/imdm/12.0?topic=mdm-consent-management

[7] P. V. Kakarlapudi and Q. H. Mahmoud, "Design and development of a blockchain-based system for private data management," *Electronics*, vol. 10, no. 24, p. 3131, Dec. 2021, doi: 10.3390/electronics10243131.

[8] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Jul. 10, 2024. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[9] *General Data Protection Regulation (GDPR). Right to erasure ('Right to be Forgotten')*. Accessed: Jul. 10, 2024. [Online]. Available: https://gdpr-info.eu/art-17-gdpr

[10] C. Lambrinoudakis, "The general data protection regulation (GDPR) era: Ten steps for compliance of data processors and data controllers," in *Trust, Privacy and Security in Digital Business* (Lecture Notes in Computer Science), vol. 11033, S. Furnell, H. Mouratidis, and G. Pernul, Eds., Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-98385-1_1.

[11] L. Sion, P. Dewitte, D. Van Landuyt, K. Wuyts, I. Emanuilov, P. Valcke, and W. Joosen, "An architectural view for data protection by design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Mar. 2019, pp. 11–20.

[12] Gartner. *Consent Management*. Accessed: Jul. 10, 2024. [Online]. Available: https://www.gartner.com/en/information-technology/glossary/consent-management

[13] European Commission. *Who Does the Data Protection Law Apply to*. Accessed: Jul. 10, 2024. [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/reform/rules-business-and-organisations/application-regulation/who-does-data-protection-law-apply_en

[14] M. Swan, *Blockchain: Blueprint for a New Economy*, 1st ed., Sebastopol, CA, USA: O'Reilly Media, 2015.

[15] W. Mougayar, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. Hoboken, NJ, USA: Wiley, 2016.

[16] A. Alkhateeb, C. Catal, G. Kar, and A. Mishra, "Hybrid blockchain platforms for the Internet of Things (IoT): A systematic literature review," *Sensors*, vol. 22, no. 4, p. 1304, Feb. 2022, doi: 10.3390/s22041304.

[17] S. B. Far, A. I. Rad, and M. R. Asaar, "Blockchain and its derived technologies shape the future generation of digital businesses: A focus on decentralized finance and the metaverse," *Data Sci. Manage.*, vol. 6, no. 3, pp. 183–197, Sep. 2023, doi: 10.1016/j.dsm.2023.06.002.

[18] S. Daoudagh, E. Marchetti, V. Savarino, R. Di Bernardo, and M. Alessi, "How to improve the GDPR compliance through consent management and access control," in *Proc. 7th Int. Conf. Inf. Syst. Secur. Privacy*, 2021, pp. 534–541.

[19] M. M. Merlec, Y. K. Lee, S.-P. Hong, and H. P. In, "A smart contract-based dynamic consent management system for personal data usage under GDPR," *Sensors*, vol. 21, no. 23, p. 7994, Nov. 2021, doi: 10.3390/s21237994.

[20] K. Rantos, G. Drosatos, A. Kritsas, C. Ilioudis, A. Papanikolaou, and A. P. Filippidis, "A blockchain-based platform for consent management of personal data processing in the IoT ecosystem," *Secur. Commun. Netw.*, vol. 2019, pp. 1–15, Oct. 2019, doi: 10.1155/2019/1431578.

[21] V. Jaiman and V. Urovi, "A consent model for blockchain-based health data sharing platforms," *IEEE Access*, vol. 8, pp. 143734–143745, 2020, doi: 10.1109/ACCESS.2020.3014565.

[22] D. Tith, J.-S. Lee, H. Suzuki, W. M. A. B. Wijesundara, N. Taira, T. Obi, and N. Ohyama, "Patient consent management by a purpose-based consent model for electronic health record based on blockchain technology," *Healthcare Informat. Res.*, vol. 26, no. 4, pp. 265–273, Oct. 2020, doi: 10.4258/hir.2020.26.4.265.

[23] S. Barbaria, M. C. Mont, E. Ghadafi, H. M. Machraoui, and H. B. Rahmouni, "Leveraging patient information sharing using blockchain-based distributed networks," *IEEE Access*, vol. 10, pp. 106334–106351, 2022, doi: 10.1109/ACCESS.2022.3206046.

[24] T. M. Kim, S.-J. Lee, D.-J. Chang, J. Koo, T. Kim, K.-H. Yoon, and I.-Y. Choi, "DynamiChain: Development of medical blockchain ecosystem based on dynamic consent system," *Appl. Sci.*, vol. 11, no. 4, p. 1612, Feb. 2021, doi: 10.3390/app11041612.

[25] C. C. Agbo and Q. H. Mahmoud, "Design and implementation of a blockchain-based E-Health consent management framework," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Toronto, ON, Canada, Oct. 2020, pp. 812–817, doi: 10.1109/SMC42975.2020.9283203.

[26] L. D. Costa, B. Pinheiro, W. Cordeiro, R. Araújo, and A. Abelém, "Sec-health: A blockchain-based protocol for securing health records," *IEEE Access*, vol. 11, pp. 16605–16620, 2023, doi: 10.1109/ACCESS.2023.3245046.

[27] G. Albanese, J.-P. Calbimonte, M. Schumacher, and D. Calvaresi, "Dynamic consent management for clinical trials via private blockchain technology," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 11, pp. 4909–4926, Nov. 2020, doi: 10.1007/s12652-020-01761-1.

[28] F. M. V. Filho, B. L. D. A. Batista, J. C. Júnior, and J. N. De Souza, "Heimdall: Blockchain-based consent management framework," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Bangkok, Thailand, Sep. 2023, pp. 487–492, doi: 10.1109/ICOIN56518.2023.10048920.

[29] I. Román-Martínez, J. Calvillo-Arbizu, V. J. Mayor-Gallego, G. Madinabeitia-Luque, A. J. Estepa-Alonso, and R. M. Estepa-Alonso, "Blockchain-based service-oriented architecture for consent management, access control, and auditing," *IEEE Access*, vol. 11, pp. 12727–12741, 2023, doi: 10.1109/ACCESS.2023.3242605.

[30] A. B. Haque, A. K. M. N. Islam, S. Hyrynsalmi, B. Naqvi, and K. Smolander, "GDPR compliant blockchains—A systematic literature review," *IEEE Access*, vol. 9, pp. 50593–50606, 2021, doi: 10.1109/ACCESS.2021.3069877.

[31] S. Han and S. Park, "A gap between blockchain and general data protection regulation: A systematic review," *IEEE Access*, vol. 10, pp. 103888–103905, 2022, doi: 10.1109/ACCESS.2022.3210110.

[32] H. Ulusoy, M. Kantarcioglu, E. Pattuk, and L. Kagal, "AccountableMR: Toward accountable MapReduce systems," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Santa Clara, CA, USA, Oct. 2015, pp. 451–460, doi: 10.1109/BIGDATA.2015.7363786.

[33] *Hyperledger Fabric*. Accessed: Jul. 10, 2024. [Online]. Available: https://www.hyperledger.org/projects/fabric

[34] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher, and S. McLachlan, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *J. Amer. Med. Inform. Assoc.*, vol. 25, no. 3, pp. 230–238, Mar. 2018, doi: 10.1093/jamia/ocx079.

[35] The MITRE Corporation. *SyntheticMass*. Accessed: Jul. 10, 2024. [Online]. Available: https://synthea.mitre.org/downloads

OZGU CAN (Member, IEEE) received the B.S. degree in computer engineering from Selçuk University, Konya, Türkiye, the M.S. degree from the International Computer Institute, Ege University, Izmir, Türkiye, and the Ph.D. degree in computer engineering from the Department of Computer Engineering, Ege University.

From 2011 to 2013, she was a Visiting Researcher with the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas (UTD), Dallas, TX, USA. She is currently an Associate Professor (Doctor) with the Department of Computer Engineering, Ege University. Her research interests include information security, data privacy, privacy-preserving techniques, knowledge engineering, and semantic web technologies.

TUNAHAN DAG received the B.Sc. degree in computer engineering from Izmir Institute of Technology, Türkiye, in 2019, and the M.Sc. degree in computer engineering from Ege University, Türkiye, in 2023. He is currently a Blockchain Architect with GoArt, a leading Web3 Company, Türkiye. His research interests include blockchain technologies, cryptography, consent management, and decentralized systems.

MURAT KANTARCIOGLU (Fellow, IEEE) is currently an Ashbel Smith Professor with the Computer Science Department and the Director of the Data Security and Privacy Laboratory, The University of Texas at Dallas (UTD). He is also a Faculty Associate with the Harvard Data Privacy Laboratory and a Visiting Scholar with the RISE Laboratories, UC Berkeley. His research interests include the integration of cyber security, data science, and blockchains for creating technologies that can efficiently and securely share and analyze data.

Dr. Kantarcioglu is also a fellow of American Association for the Advancement of Science (AAAS) and a Distinguished Member of ACM. He was a recipient of various awards, including NSF CAREER Award, American Medical Informatics Association (AMIA) 2014, Homer R. Warner Award and the Institute of Electrical and Electronics Engineers (IEEE), and Intelligence and Security Informatics (ISI) 2017 Technical Achievement Award for his research in data security and privacy.