SecureConsent: A Blockchain-based Dynamic and Secure Consent Management for Genomic Data Sharing

***Abstract*—The potential of precision oncology initiatives heavily relies on sharing and analyzing genomic data across diverse patient groups. However, the sensitivity of genomic data raises concerns about consent, and data control, hindering patient participation in such initiatives. Existing healthcare data-sharing methods are unable to fully address these issues, and they also do not take into account patient preferences and requirements within their models. Therefore in this paper, we introduce "SecureConsent", a Patient-Centric consent management system designed for sharing genomic data. SecureConsent caters to patient preferences, allowing them to have control of how their genomic data is shared and used within precision oncology initiatives. It enables patients to modify their consent at any moment through the implementation of dynamic consent. Moreover, it integrates a decentralized authentication and access control mechanism to establish a robust and patient-centric framework. SecureConsent also presents a user-friendly interface for simple interaction between patients and those requesting data. We conduct a performance evaluation of our blockchain-based model to establish its system efficiency, which includes analyzing gas cost, read latency, read throughput, transaction delay, and transaction throughput.**

***Index Terms*—Consent Management, Access control, Blockchain, Genomic Data**

I. Introduction

Precision oncology has emerged as a game-changer in cancer treatment therapy by tailoring medical care to each patient's unique characteristics and disease type [1]–[3]. It involves genomic profiling to gain detailed insights into the genetic composition and biological mechanisms of a patient's tumour [4]. This enables healthcare practitioners to make better treatment decisions and select medications that are more likely to be beneficial while minimizing potential negative effects. In order for patients to truly benefit from precision oncology initiatives, the system must analyze genomic data from diverse sets of patients to reveal valuable patterns and correlations. AI and machine learning algorithms can then be applied to this diverse dataset to find new patterns, correlations, and insights in the diverse information [5]–[7]. This underscores the significance of sharing genomic data within a learning healthcare system.

To achieve responsible genetic data sharing, it is essential to address several critical challenges that current healthcare data management solutions face. Current systems do not allow data to be shared and processed outside the genomic laboratories. Genomic data is stored in centralized repositories creating data silos that restrict outside access. To analyze and utilize data outside its original domain it is necessary to provide an interoperable platform. Although facilitating data sharing is crucial, consent management and data protection are critical concerns for patients [8]. Existing data-sharing platforms often lack robust consent management. Patients' control over their data is not absolute. They are unable to decide who can access their data and for what reasons. While informed consent management solutions exist, a mechanism for dynamically managing consent with patient-centric access control specifically tailored for the sharing of genomic data is lacking. This necessitates the need for patient-centric genomic data-sharing that can incorporate dynamic consent which allows patients to specify their data-sharing preferences, grant or revoke access to their data, track data usage, and update consent choices.

Blockchain technology [9], [10] offers a range of features including, decentralization, accountability and immutability that can be harnessed to enhance the integrity, discoverability, and accessibility of genomic data. Smart contracts [11] can be leveraged to implement a secure and accountable data governance model facilitating a dynamic consent model. While blockchain has found applications for developing access control in healthcare [12]–[14] for data sharing, it has also been used by several researchers in implementing consent management [15]–[19]. Nevertheless, a comprehensive solution for patient-centric access control and dynamic consent management in the context of genomic data sharing, specifically addressing the data sharing preferences of cancer patients is currently lacking.

This paper addresses this gap by introducing SecureConsent: blockchain-based dynamic and secure consent management for genomic data sharing for cancer patients. The SecureConsent allows the patients to manage their consent followed by an access control using blockchain technology. The following are the three major contributions of this paper:

1) The requirements are formulated through focus group sessions conducted on cancer patients. The requirements presented include patient-controlled access, dynamic consent management, a comprehensive data access request process, transparent data transactions, and interactive communication for user engagement and insights.
2) The SecureConsent architecture is presented that implements dynamic consent and establishes a resilient, patient-centric framework with decentralized authentication and access control. The architecture is supported by

a user-friendly web interface, enabling patients to manage sharing of genomic data easily.

3) The performance of the proposed hybrid blockchain solution is evaluated and analysed. The performance is evaluated in terms of gas cost, read latency, read throughput, transaction delay, and transaction throughput.

The paper is organized as follows: the related work has been given in Section II. In Section III, the proposed architecture is presented along with the requirements and user interface. Section IV presents the performance analysis of the blockchain system. Finally, Section V concludes this paper.

II. Related Work

Recent technological advancements have significantly reduced the cost of genome sequencing resulting in an explosion of genetic data that is essential for many research endeavors [8]. Nonetheless, the advancement of genomics research depends on the discoverability, accessibility, and availability of genetic data. [20]. Current centralized repositories have limited access and lack interoperability. Few attempts have been made to facilitate genomic data sharing across different data centres. Two prominent efforts include the GA4H Beacon project [21] and Dicipher [22]. The GA4GH Beacon Project worked on connecting diverse genomic databases across institutions to enable queries related to genetic data. Whereas Dicpher introduced an intermediary data-sharing system to enable controlled collaboration between data centres. However, these solutions lack mechanisms to actively involve patient consent in managing and sharing their genetic data. Additionally, they do not integrate a decentralized governance structure, resulting in a centralized access control system.

Decentralized and distributed technologies have been suggested as a potential solution to promote genomic data sharing [23]. Blockchain technology, renowned for its decentralised and immutable nature, operates as a ledger comprising transactions grouped into blocks. In the realm of blockchain-based healthcare data-sharing systems, various innovative frameworks have been proposed. In Healthchain [12] an electronic health record system is introduced that uses chain codes on Hyperledger Fabric for data exchange and access management. The health data is encrypted and kept off-chain in IPFS, with only their hash values stored in the blockchain. MedRec [13] provides a decentralized EHR management system using Ethereum smart contracts to provide authentication, confidentiality, accountability and data sharing crucial considerations when handling sensitive information. DSMAC, as detailed in [14], employs a blockchain-based self-sovereign approach to enable decentralized self-management of data access. The DSMAC system harnesses smart contracts to enforce role-based access control policies. Although these solutions facilitate patient control they lack dynamic consent allowing patients to change the access to their data over time.

Few efforts have been made to implement consent management using blockchain data for healthcare data. For instance, the paper in [15] introduces a blockchain-based consent management system for personal fitness data. The solution aims to mitigate privacy issues while allowing healthcare providers to access the fitness data of patients. [18] introduces a blockchain-based e-health consent management framework that aims to make it more transparent and auditable. Dwarna is presented in [16] that harnesses blockchain to enable dynamic consent in biobanking. It enhances trust in biobanking by providing research partners with control over study participation, consent withdrawal, and data destruction requests. Additionally, a blockchain-backed service-oriented architecture was proposed [17] to ensure GDPR-compliant execution of patient consents. Nonetheless, these blockchain-based solutions lack user-centric access control and authorisation. Healthcare consent management systems can be made more transparent and robust in terms of data protection by incorporating user-centric access control methods. Furthermore, existing solutions based on blockchain, often overlook the integration of patient preferences in the design of consent management and access control.

There is a research gap in successfully integrating user-centric access control and dynamic permission management for genomic data sharing. Existing blockchain-based efforts either focus on implementing data authorization for medical data or on building a consent management system. They fail to offer a comprehensive framework that includes both robust user-centric access control systems and dynamic consent mechanisms that allow users to adjust their data-sharing choices over time. Moreover, none of the existing solutions cater to the requirements of cancer patients whose data is being shared. Therefore, in order to understand the demands of cancer patients for access to their data, we present the requirements based on our engagements with cancer patients on the topic of reporting, permissions, data security, data access requests, and data governance. Based on these requirements we present a comprehensive solution that includes dynamic consent management and implements authorisation for patient genomics data exchange using blockchain technology.

III. Proposed System Model

This section introduces SecureConsent, a blockchain-based dynamic consent management solution for genomic data sharing. First, we outline the requirements of our proposed system extracted from focus group studies with cancer patients. Following this, we present and describe the system architecture of SecureConsent. Lastly, we present the web application interface of SecureConsent.

A. Requirements

We derived the requirements of our proposed system from focus group studies with cancer patients. The focus group study comprised four sessions involving a total of 22 participants. All participants were adults with a history of or ongoing cancer diagnosis. The study sessions were conducted across four different locations in Canada [24]. The study explored the preferences and attributes patients with cancer would like to have for granting access to their personal genomic data for precision oncology research. Several key areas were

discussed and explored with the participants, including data governance, data security, data access requests, consent types, and reporting. The following requirements were extracted from the transcripts of the focus group sessions:

1) *Requirement 1*: *Patient-centric access control*: The system must enable patients to have control over their data access by allowing them to grant or deny access as per their preferences. This control includes robust authentication and authorization mechanisms to ensure secure and authorized data access.
2) *Requirement 2*: *Data access request (DAR)*: The system should enable patients to make informed decisions about their data access requests (DAR). The DAR should encompass requester information and the data utilization plan. The data utilization plan should include the purpose of access, expected results, ethical approval, and data access duration.
3) *Requirement 3*: *Dynamic consent*: The system must implement dynamic consent allowing patients to modify their consent over time, including the ability to revoke previously granted data access or change the period of the data access.
4) *Requirement 4*: *Data transactional transparency*: The system must maintain a detailed immutable history that enables patients to review and track all instances of data access for accountability. This provides patients with a clear view of genomic data transfer, the recipients of their data, and whether the data has been shared in compliance with the granted consent.
5) *Requirement 5*: *Interactive communication*: The system should establish robust communication between patients and data requestors. This entails granting users the capability to pose specific questions to data requestors and allowing them to view periodic traceability reports of their data that contain comprehensive insights into data handling and usage.

The resulting requirements are designed to prioritize patient-centricity, ensuring that individuals have control over access to their data, with a focus on security, transparency, and dynamic consent while facilitating interactive communication.

B. System Architecture

We utilized the DLT question-led system design framework, as outlined in [25], to define our purpose, goals, functionalities, and system entities based on the earlier established requirements. The SecureConsent architecture consists of five entities including Data Owner (*DO*), Data Custodians (*DC*), Data Requestor (*DR*), Blockchain and Data Storage. The *DO* represents the cancer patient to whom the genomic data is related. The *DC* represents institutes such as hospitals and testing laboratories that are responsible for collecting, storing and safeguarding patient data in secure data repositories. The *DR* represents organizations such as research institutes and pharmaceutical companies that seek access to specific genetic data.

SecureConsent harnesses the blockchain to ensure security, decentralization, and accountability. A private consortium blockchain managed by trusted parties is used to implement authentication, authorization and consent management. A public blockchain is used to guard the system against unauthorized alterations and potential attacks from the consortium partners. The Data Storage is used to store genomic and other required data. Each *DC* has its own private genomic data repository that adheres to high-security standards. Furthermore, a decentralized file storage system, InterPlanetary File System(IPFS), is used to store the data utilization policy for the data access requests sent.

The system architecture of SecureConsent is illustrated in Figure 1. The platform accommodates three types of users as previously defined. Every user utilizes a digital wallet, which has a public and private key combination. The public key is used to generate the public address, while the private key of the user is kept in the wallet and is used to sign transactions. The public address serves as a pseudonymous identity on the blockchain. To ensure secure data access, the data access requests are encrypted and stored on blockchain, allowing only authorized individuals to use them. The data utilization policy, on the other hand is stored off-chain on IPFS to avoid potential identification of users, given the sensitive nature of the information included in the policy.

Now, let's examine the steps of how SecureConsent operates as illustrated in Figure 1:

Step 1 Data Identifier Upload: The *DC* securely stores patient genomic data into a genetic repository and records data file identifiers on the consortium blockchain.

Step 2 Upload Data Utilization Policy: The *DR* uploads the Data Utilization Policy to IPFS that includes the purpose of access, expected results, ethical approval, and data access duration.

Step 3 Data Access Request: The *DR* initiates a data access request using the data identifiers, encrypting the request with the *DO*'s public key, and stores it on the blockchain. The request also includes the data utilization policy link

Step 4 Retrieve Data Utilization Policy : The *DO* use their private key to decrypt the data access request. The data access request contains the information regarding the *DR*. Moreover, the *DO* uses the IPFS link to retrieve data utilization policy.

Step 5 Consent Response: The *DO* records approval or rejection on the blockchain. Upon approval, *DO* generates a cryptographically signed JWT access token with specified access duration, encrypted with the *DR*'s public key, and uploads it to the blockchain.

Step 6 Access Token Retrieval: The *DR* retrieves and decrypts the access token with their private key. The access token contains the access details and the IPFS link to *DO*'s identity along with the decryption key.

Step 7 Access Token Transfer: The *DR* sends the token to the *DC* to access the genetic file from the repository.

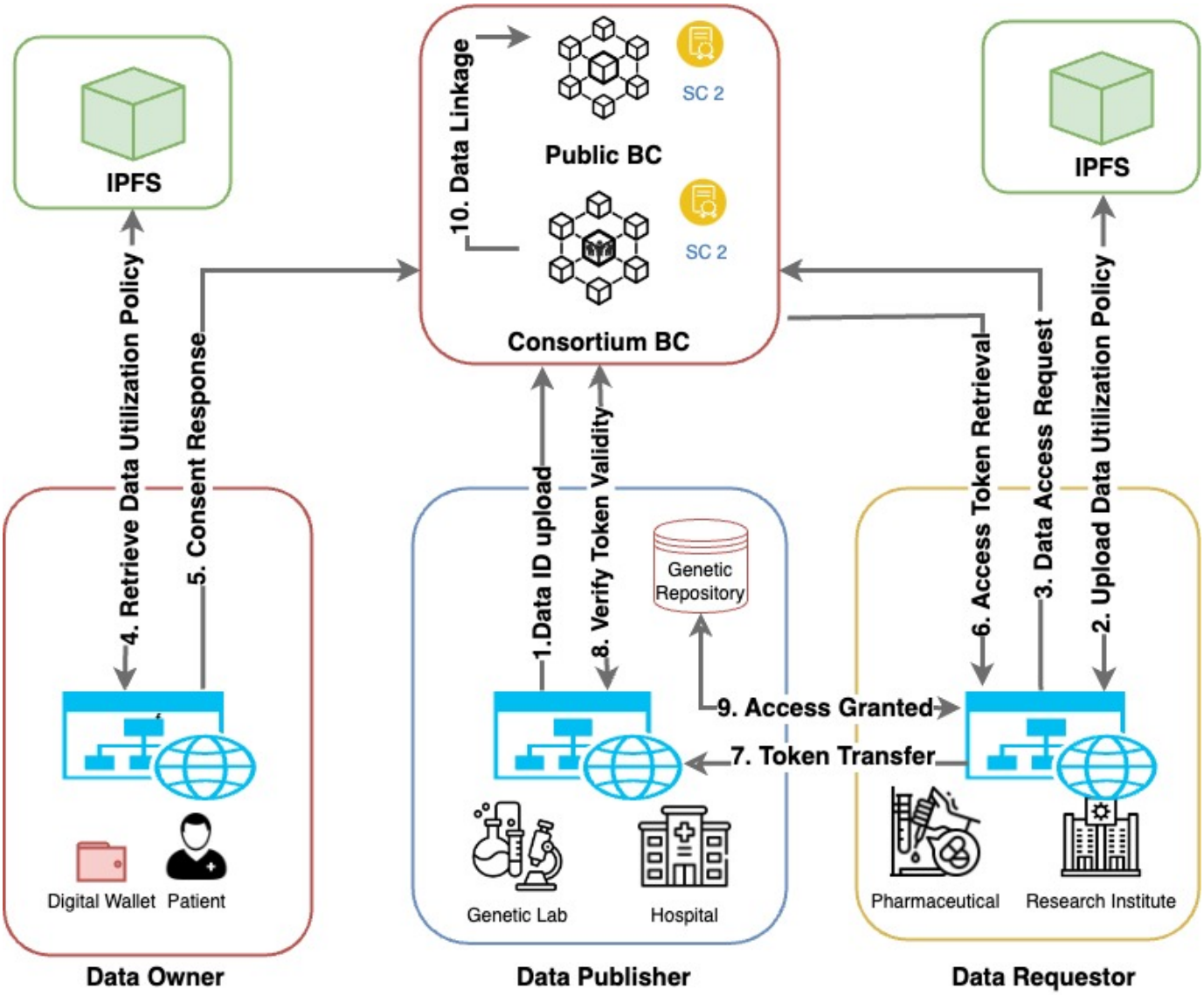

Fig. 1. SecureConsent System Architecture

Step 8 Verify Token Validity: The *DC* examines the access token, verifying the *DO*'s signature, checking for token revocation on the blockchain, and confirming the *DO* permissions.

Step 9 Access Granted: If the token verification is successful the genomic data access is granted until its expiry or it is revoked.

Step 10 Data Finalize: The hash of each block added to the consortium blockchain is uploaded to a public blockchain for enhanced security and verification.

SecureConsent implements a secure and dynamic consent management system using smart contracts, access tokens, and cryptography. SecureConsent allows patients to grant or reject access, specifying the access expiry and revoking the access later on if needed. Additionally, it enables patients to change the access duration, which calls for the revoking of the old access token and the issuing of a new one with the updated expiry time. SecureConsent implements JSON Web Tokens (JWTs) for data access. The data owner creates the JWT that includes the data owner address, data requestor address, data access link, data identifier, data access expiry, identity link, the decryption key for identity and digital signature. The JWT can be used to access the specified data from the data repository. By examining the signatures and cross-referencing the addresses from them with the blockchain, the legitimacy of the access token and access permissions can be confirmed. The access is granted until the expiry of the token. If there is ever a need for the patient to revoke access or change access duration, it is efficiently managed through updates on the blockchain.

SecureConsent employs two smart contracts $SC1$ and $SC2$ as shown in Figure 1. While $SC2$ is deployed on the public blockchain, $SC1$ functions on the consortium blockchain. The

$SC1$ is used to register, authenticate, verify and attest users and implement the access control and consent management functions. $SC2$ is used to link data blocks from the consortium blockchain to the public blockchain for added security. The SecureConsent system is reliant on the following set of smart contract functions:

- User Registration Functions: Users can register, authenticate, and get identity attestations with the `RegisterID`, `VerifyID`, and `AttestID` functions.
- Data Management Functions: The `addFileID` function is utilized by *DC* to upload data identifiers, making them available for viewing and requesting. The `hideFile` function enables data owners to hide data identifiers they do not want to be requested.
- Consent Management Functions: The `addRequest` function is used by *DR* to submit data access requests including the IPFS link for the data access details. Data owners use the `respondRequest` function to approve or reject incoming access requests. Additionally, the `RevokeAccess` function allows data owners to revoke previously granted access, while the `modifyAccess` function enables *DO* to both revoke previously granted access and upload a new response with updated access details.
- Project Management Functions: The `addQuestion` and `answerQuestion` functions facilitate communication between *DR* and *DO*, allowing them to exchange information and clarify details related to data access and data usage. The `addReporting` and `viewReporting` functions allow the periodic data utilization reports to be uploaded and viewed.
- Blockchain Integration Functions: Trusted consortium participants can register as public organizations using the `RegisterConsortium` function on SC2, while consortium partners can upload the hash of each mined block to the public blockchain using the `UploadBlockHash` function.

C. Web Application Interface

SecureConsent provides a user-friendly interface that is suited to each user type. Users are required to authenticate with a digital wallet before using the application. The wallet address is used as a pseudonymous identifier to represent each user on the blockchain network. The wallet address allows the application to determine the type of user and then enable access to specific web pages. Figure 2 illustrates the web application interface intended for patients. This interface is divided into three components including data repository, consent management, and activity log.

The data repository section is dedicated to managing the different genomic data files of each patient. The data listing is presented in a structured tabular format, displaying information such as data ID, data type, data custodian, upload date, and upload time. Using the action column the patients can effortlessly access, view, and choose to hide their genetic data, thereby controlling access to their data. The consent management section empowers patients to handle data access requests. Patients can review details such as the requestor name and access details, which encompass expected results, ethical approval details, regulatory compliance, feedback, and reporting period, enabling informed consent. Patients can use the action column to grant or deny access, establish time limitations for data access, and revoke data access.

The activity log section gives patients access to the transaction history of their genetic data interactions by recording and presenting significant events such as data uploads, access requests, grants, denials, and revocations. These elements are seamlessly combined on the patient web page to provide a user-centric platform for sharing and managing genetic data that is customized to patient preferences based on research conducted in focus groups. Patients can effectively manage their information, restrict access, and keep an extensive activity log. This website essentially serves as a key component of our decentralized genomic data-sharing platform, promoting openness and ownership over genetic data while guaranteeing safe, patient-led data management.

IV. Performance Evaluation

In this section, we will look at the implementation and performance of SecureConsent. SecureConsent employs a hybrid strategy, using a consortium blockchain with the public blockchain. First, we will go over the implementation setup, highlight the important parameters, and describe the performance metrics. Following that, we compute and analyze the performance of our blockchain solution in terms of *Gas*, *Latency*, and *Transaction Throughput*.

A. Setup

Our proposed solution is implemented using a private hyperledger Besu network version 22.4. We used the Quorum Developer Quickstart using docker images to deploy blockchain nodes. We choose the Quorum Developer Quickstart to deploy besu nodes, which includes vital functions and endpoints for node communication. Our blockchain network consisted of four Besu nodes running in Docker containers (version 20.10.14) on Ubuntu 20.04-powered virtual machines with 20 GB RAM. We configured a custom genesis file to initialize the blockchain, setting the ”gasLimit” to a maximum value of ”0x1fffffffffffff” and used IBFT consensus. We linked the Remix IDE to the Besu Blockchain using the Metamask wallet to make smart contract implementation and interactions easier. Hyperledger Caliper version 0.4.2 was utilized to establish a connection and evaluate the performance of the implemented Besu blockchain networks.

We investigate the blockchain performance deployed in SecureConsent using three important metrics: *Gas*, *Latency*, and *Transaction Throughput*. *Gas* is computed for smart contract functions implemented on private and public blockchains. Meanwhile, *Latency* and *Transaction Throughput* are analyzed specifically for the performance of private blockchain. *Gas* refers to the computing resources necessary to execute transactions and smart contracts. We compute the *Gas* utilized

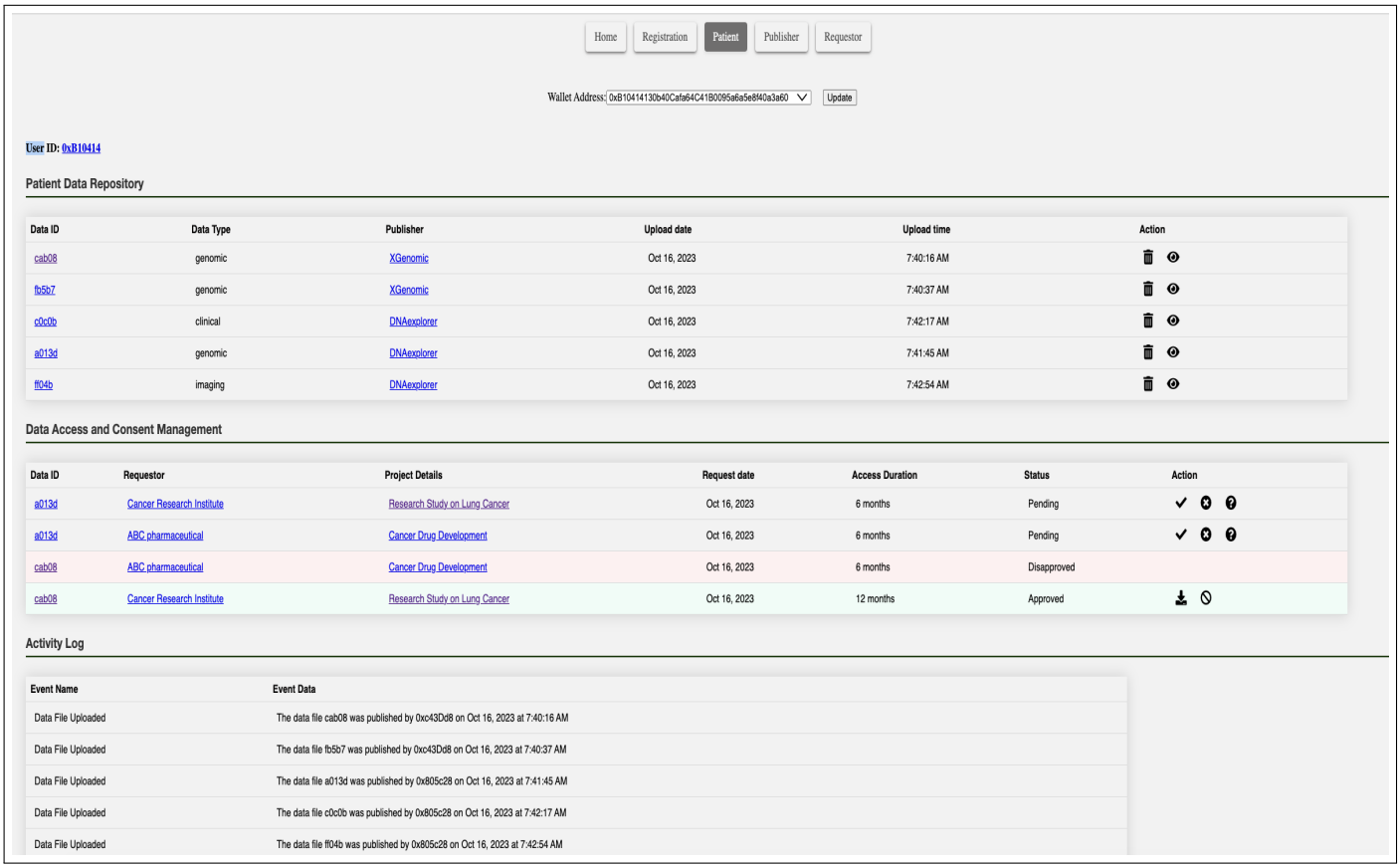

Fig. 2. Patient interface of SecureConsent Application

for each smart contract function on the private blockchain to demonstrate computational efficiency and resource utilization. For public blockchain, we utilized *Gas* to compute the gas cost spent on various EVM machines. The *Transaction Throughput* refers to the number of transactions that the blockchain can verify in a second. It is represented by Transactions Per Second (TPS) which is a vital indicator of the network's capability to handle a high volume of transactions within a given time frame. *Latency* refers to the amount of time required to validate a single transaction. *Latency* aids in understanding the blockchain's capacity to handle transactions ensuring that network infrastructure is appropriately scaled to meet user needs.

B. Results

We assess SecureConsent's performance by first estimating the gas required for the primary operations of $SC1$ in the private blockchain. We then compute the throughput and latency of the transactions to evaluate the efficiency and responsiveness of the private blockchain network. We also expand our evaluation to public blockchains, comparing the gas prices of different EVM-based platforms and calculating the transaction costs of implementing $SC2$ on different public blockchain infrastructures.

Table I presents an overview of gas utilized with several core functions of SecureConsent $SC1$. Each function has its own gas cost, which reflects the computing resources required for its execution. `addFileID` has the largest gas cost, owing to its involvement in establishing a data repository for patient files. This function involves significant data storage, which contributes to the higher gas costs. `RegisterID` and `addRequest` are computationally expensive functions as they require resource-intensive operations to register user registration and create data access requests respectively. On the other hand, functions like `VerifyID`, `AttestID`, `respondRequest`, and hideID consume less gas as they require fewer storage operations. `addQuestion`, `answerQuestion addReporting`, `viewReporting` are examples of communication-facilitating functions that also maintain relatively reduced gas costs due to their simple logic and minimal computational steps.

The private blockchain's *Transaction Throughput* is displayed in Table II. It illustrates how the volume of transactions submitted has an impact on the blockchain's processing speed per second. Increased transaction frequency seems to positively affect throughput, indicating the scalability and effectiveness of the blockchain infrastructure. It is found that the blockchain system may achieve an optimal throughput of up to 50 TPS. A decrease in throughput is seen when the number of transactions per transaction surpasses 50. The declining trend in throughput beyond 50 transactions per second indicates

TABLE I
SMART CONTRACT FUNCTIONS WITH GAS COST

Function	Gas Cost
`RegisterID`	142204
`VerifyID`	19362
`AttestID`	24599
`addFileID`	248440
`hideFile`	14588
`addRequest`	39927
`respondRequest`	26099
`RevokeAccess`	27022
`modifyAccess`	28011
`addQuestion`	26099
`answerQuestion`	24011
`addReporting`	23099
`viewReporting`	22024

that the system would not be able to continue operating at peak efficiency with more transactions. Table III illustrates the *Latency* which shows that the time required to complete a transaction decreases with transaction frequency. The average *Latency* is always less than one second for frequencies lower than 50. However, the reaction time substantially increases, reaching up to 5 seconds, when the frequency climbs over 50. This finding emphasizes how transaction frequency affects processing efficiency and emphasizes the necessity for careful thought and improvement in situations with increased transaction volumes. More investigation and improvement are needed to solve this throughput decline and increased latency at higher frequencies, especially in scenarios with greater transaction rates. Consensus protocols like Fast-HotStuff [26] can be used to achieve much higher *Latency* and *Transaction Throughput*.

TABLE II
BLOCKCHAIN PERFORMANCE IN TERMS OF TRANSACTION THROUGHPUT

Frequency (tx/s)	Throughput
1	1
10	10
50	50
100	93
500	470
1000	959

TABLE III
BLOCKCHAIN PERFORMANCE IN TERMS OF LATENCY

Frequency (tx/s)	Min Latency	Max Latency	Average Latency
1	0.015	0.022	0.0185
10	0.021	0.033	0.027
50	0.93	1.12	1.025
100	2.83	3.97	3.4
500	3.41	5.56	4.485

Table IV provides a comparative analysis of gas costs across various public EVM-based blockchains. The gas used for adding the block hash and the block number is standardized at $23,824$ gas. The average gas price and token price for each blockchain token are considered as of October 14, 2023. Notably, Ethereum emerges with the highest gas cost of USD 0.26, marking it as the most expensive option. Layer 2 solutions on the Ethereum network, such as Polygon, Arbitrum, and Optimism, provide significant cost savings. Notably, Arbitrum and Optimism provide lower gas prices, while Polygon, operating as a side chain, emerges as the most cost-effective solution. Avalanche has one of the highest gas prices among other Layer 1 blockchains, while Fantom has one of the lowest gas costs even with a higher gas price. Harmony performs exceptionally well, making it the most economical option in our research.

V. CONCLUSION AND FUTURE WORK

In this paper, we present SecureConsent, a blockchain-based dynamic consent management framework for precision oncology's genomic data sharing. We aim to provide a patient-centric solution that empowers individuals with control over access to their data, addressing challenges related to data control and data protection. SecureConsent utilizes a consortium blockchain, integrating private and public blockchains, and employs smart contracts to ensure secure and transparent data management, dynamic consent, and project-based access control. Performance evaluations reveal the system's capability to handle diverse transactions with low gas costs, making it a practical solution for genomic data sharing. SecureConsent has the potential to significantly impact precision oncology and genomic research by enabling patient-controlled, secure, and transparent data sharing.

While SecureConsent represents a promising solution towards patient-centric control over genomic data in precision oncology, it is essential to acknowledge that it assumes a level of trust between patients and entities requesting access to their genomic data. However, for certain patients trusting third parties to store and process their genomic data can be a barrier. It will be important to solve this constraint in order to achieve wider acceptance and adoption. Therefore, in our future work, we intend to ensure patient data privacy by incorporating Trusted Execution Environments (TEEs) into the SecureConsent architecture. The genomic data-sharing procedure will have an additional degree of privacy by processing the genomic data in TEE. We also intend to explore Zero Knowledge Proof to prove the authenticity of user identity information without revealing the information itself. By integrating TEEs and ZKPs, we aim to make SecureConsent more resilient, privacy-focused, and secure in precision oncology's genomic data sharing.

REFERENCES

[1] J. Mateo, L. Steuten, P. Aftimos, and et al., "Delivering precision oncology to patients with cancer," *Nature Medicine*, vol. 28, pp. 658–665, 2022.

[2] S.-Y. Ku, M. E. Gleave, and H. Beltran, "Towards precision oncology in advanced prostate cancer," *Nature Reviews Urology*, vol. 16, pp. 645–654, 2019. Accepted 10 September 2019, Published 07 October 2019, Issue Date November 2019.

[3] U. N. Lassen, L. E. Makaroff, A. Stenzinger, A. Italiano, G. Vassal, J. Garcia-Foncillas, and B. Avouac, "Precision oncology: a clinical and patient perspective," *Future Oncology*, vol. 17, no. 30, pp. 3995–4009, 2021.

[4] L. R. Yates and et al., "The european society for medical oncology (esmo) precision medicine glossary," *Annals of Oncology*, vol. 29, pp. 30–35, 2018.

Public EVM Blockchain	Gas Price (Gwei)	Gas Used	Token Price in USD	Gas Cost in USD
Ethereum (ETH)	7.1	0.00017	1,554.04	0.26484568
Arbitrum (ETH)	0.1	0.000002	1,554.04	0.00310808
Optimism (ETH)	0.1	0.000002	1,554.04	0.00310808
Polygon (MATIC)	56	0.001338	0.5185	0.00069123
Fantom (FTM)	70.8	0.001696	0.1858	0.00031496
Avalanche (AVAX)	25.3	0.000606	9.21	0.00562526
Harmony (ONE)	101	0.002415	0.009216	2.22654E-05

TABLE IV
GAS COSTS FOR EVM-BASED BLOCKCHAINS AS OF $14Oct2023$

[5] Z. Dlamini, F. Z. Francies, R. Hull, and R. Marima, "Artificial intelligence (ai) and big data in cancer and precision oncology," *Computational and structural biotechnology journal*, vol. 18, pp. 2300–2311, 2020.

[6] L. Wei, D. Niraula, E. D. Gates, J. Fu, Y. Luo, M. J. Nyflot, S. R. Bowen, I. M. El Naqa, and S. Cui, "Artificial intelligence (ai) and machine learning (ml) in precision oncology: A review on enhancing discoverability through multiomics integration," *The British Journal of Radiology*, vol. 96, no. 1150, p. 20230211, 2023.

[7] S. Bhalla and A. Laganà, "Artificial intelligence for precision oncology," *Computational methods for precision oncology*, pp. 249–268, 2022.

[8] L. Bonomi, Y. Huang, and L. Ohno-Machado, "Privacy challenges and research opportunities for genomic data sharing," *Nat Genet*, vol. 52, pp. 646–654, 2020.

[9] A. S. Rajasekaran, M. Azees, and F. Al-Turjman, "A comprehensive survey on blockchain technology," *Sustainable Energy Technologies and Assessments*, vol. 52, p. 102039, 2022.

[10] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain: research and applications*, vol. 3, no. 2, p. 100067, 2022.

[11] P. Sharma, R. Jindal, and M. D. Borah, "A review of smart contract-based platforms, applications, and challenges," *Cluster Computing*, vol. 26, no. 1, pp. 395–421, 2023.

[12] S. Chenthara, K. Ahmed, H. Wang, F. Whittaker, and Z. Chen, "Healthchain: A novel framework on privacy preservation of electronic health records using blockchain technology," *PLOS ONE*, vol. 15, no. 12, p. e0243043, 2020.

[13] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, pp. 25–30, 2016.

[14] H. Saidi, N. Labraoui, A. A. A. Ari, L. A. Maglaras, and J. H. M. Emati, "Dsmac: Privacy-aware decentralized self-management of data access control based on blockchain for health data," *IEEE Access*, vol. 10, pp. 101011–101028, 2022.

[15] M. Alhajri, C. Rudolph, and A. S. Shahraki, "A blockchain-based consent mechanism for access to fitness data in the healthcare context," *IEEE Access*, vol. 10, pp. 22960–22979, 2022.

[16] N. Mamo, G. Martin, M. Desira, and et al., "Dwarna: A blockchain solution for dynamic consent in biobanking," *Eur J Hum Genet*, vol. 28, pp. 609–626, 2020.

[17] 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.

[18] C. C. Agbo and Q. H. Mahmoud, "Design and implementation of a blockchain-based e-health consent management framework," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 812–817, 2020.

[19] K. Varshini Naik, T. Vijaya Murari, and T. Manoj, "A blockchain based patient consent management technique for electronic health record sharing," in *2022 IEEE 7th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, vol. 7, pp. 214–219, 2022.

[20] ACMG Board Of Directors, "Laboratory and clinical genomic data sharing is crucial to improving genetic health care: a position statement of the american college of medical genetics and genomics," *Genet Med*, vol. 19, no. 7, pp. 721–722, 2017.

[21] T. G. A. for Genomics and Health*, "A federated ecosystem for sharing genomic, clinical data," *Science*, vol. 352, no. 6291, pp. 1278–1280, 2016.

[22] E. Bragin, E. Chatzimichali, C. Wright, M. Hurles, H. Firth, A. Bevan, and G. Swaminathan, "Decipher: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation," *Nucleic Acids Res*, vol. 42, no. Database issue, pp. D993–D1000, 2014.

[23] M. Shabani, "Blockchain-based platforms for genomic data sharing: a de-centralized approach in response to the governance problems?," *Journal of the American Medical Informatics Association*, vol. 26, no. 1, pp. 76–80, 2019.

[24] BC Cancer Research Centre, "Canadian network for learning healthcare systems and cost-effective omics innovation (cleo)." https://www.bccrc.ca/dept/ccr/projects/canadian-network-learning-healthcare-systems-and-cost-effective-omics-innovation-cleo, Year of access. Accessed on: 2023-10-17.

[25] V. L. Lemieux and C. Feng, "Building decentralized trust," *Multidisciplinary Perspectives on the Design of Blockchains and Distributed Ledgers. Krems: Springer*, 2021.

[26] M. M. Jalalzai, C. Feng, and V. Lemieux, "Fast b4b: Fast bft for blockchains," 2021.