This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008
2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

Blockchain-based electronic health record system with patient-centred data access control

Stavros Koumpounis, Mark Perry
Department of Computer Science
Brunel University London, UK
{1946252,mark.perry}@brunel.ac.uk

***Abstract*—This study examines the application of blockchain technology in addressing the increasing prevalence of mental health issues among young adults in the UK. Despite the recognition of the potential of digital health solutions and mental health apps by healthcare organizations and professionals, many individuals are hesitant to share sensitive information due to security concerns. To address this, the study proposes utilizing blockchain technology to create a patient-centred, transparent system for data access control that adheres to the General Data Protection Regulation (GDPR) and established security protocols. The research includes the development of a proof-of-concept decentralized web application, which incorporates a modified scrum methodology, test-driven development, clearly defined security requirements, and design patterns for security and gas optimization. The study concludes with an evaluation of the application, insights from automated security analytics, and recommendations for future research.**

***Index Terms*—consensus, blockchain, concurrency, patient-centred, digital health**

I. Introduction

Neuropsychiatric conditions stand among the leading causes of disability and their occurrence has been increasing worldwide. With young adults ranking high in vulnerability, suicide stands as the second most common cause of death globally and first in the UK amongst this age group [24]. Despite the importance of addressing mental illness early, only 25% of young adults in the UK receive professional help, due to several issues acting as a barrier over access to traditional and online mental health intervention and care. These include a lack of awareness and psychoeducation, limited availability and reach of services, and the fear of stigma and discrimination [10; 26].

Several reviews outline the fact that healthcare lags behind other fields in terms of security and more specifically the security regarding patients' sensitive health information. As well as being at risk of integrity and confidentiality flaws, current healthcare management systems are vulnerable to cyber-attacks. In 2017 the WannaCry attack affected computers in 80 of the 236 NHS trusts, along with more than 250,000 computers in 150 countries [30]. According to a survey by Healthwatch England, a consumer health and social care body, 53% of the 1761 people (85% of respondents) who had heard about the WannaCry attack said it had made them less confident in the ability of the NHS to protect their confidential patient information [12]. A similar sentiment was shared in the United States, where in a related study, out of the 64.5% of respondents that were concerned about breaches of privacy and security of electronic health records (EHRs), a further 12.3% reported having withheld information from healthcare professionals due to the fear of inadequate privacy and security [1].

The NHS, however, has set out a ten-year plan to become fully digital, focusing on interoperability and giving patients control of their data, exemplifying the notion of patient-centred care [23]. This, although requiring extensive reengineering of legacy systems, would increase security standards and eventually allow patients to become active agents in their own care and possibly alleviate trust barriers. The reengineering of systems would need to stay aware of legal restrictions, such as the recently introduced General Data Protection Regulation (GDPR) and incorporate it in the design [14].

This paper aims to address the need for decentralisation and a zero-trust model by providing transparent patient-centred data access control and abiding by known security practices to protect and maintain data integrity and confidentiality. This indirectly attempts to increase trust in mental health applications, which in succession would reduce the fear of stigma and bolster the reach of professional help to those who need it.

We performed a literature review to identify existing works related to mental health app development and EHR systems utilising blockchain, to identify appropriate techniques and capture requirements to be adopted.

The goals of this paper are the following:

- Identify a methodology and recognised relevant practices which will address the specialised requirements of a privacy-sensitive blockchain solution.
- Design an appropriate architecture for a decentralised application interacting with off-chain cloud storage, using appropriate design practices, patterns and GDPR guidelines.
- Develop and evaluate a prototype using Solidity, and the Hardhat development environment for the blockchain access control component, adhering to design patterns specified.

II. Background and Related Work

A (public) blockchain is part of a distributed ledger of nodes (also called 'miners') running on several different machines across the globe, forming a peer-to-peer decentralised network

This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008,2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

system. Acting like a distributed database with an immutable history of transactions or blocks (with each node holding a copy of the chain/'database'), each block is 'locked' to the previous one with the use of cryptographic hashes. Any alteration of one block would modify the hashes of all subsequent blocks and therefore produce an inconsistency when compared to the ledger of nodes which would hold the same copy.

The most popular application of blockchain technology is the Bitcoin blockchain, which facilitates decentralized financial transactions and eliminates the need for intermediaries [22]. A similar platform, Ethereum, was later introduced and allows for the execution of smart contracts on the blockchain, enabling transparent and tamper-proof storage and retrieval of information [4]. This technology is useful for creating patient-centered access control that is transparent, auditable, and respects integrity.

A current limitation around (public/permissionless) blockchains, is that nodes might be located anywhere in the world, such as that personal data would be sent outside the EEA, without the data controller knowing where, making it a struggle to comply with data transfer agreements. Patients may request for their data to be erased, which would make solutions incorporating blockchain technology more complex to define [28].

The use of blockchain technology in healthcare has gained traction in recent years due to its potential benefits for EHR management [15; 16]. However, it also has limitations, such as susceptibility to Sybil and 51% attacks on Ethereum and other proof-of-work chains [31]. Ancile suggests utilizing a permissioned blockchain to address this issue, but it sacrifices transparency of the access control layer and relies on trust in the platform managing the Ancile system [7].

FHIRChain, is a permissionless blockchain solution, designed with interoperability in mind and following the FHIR standards for record sharing [33]. Only data pointers are stored on the blockchain, using a token-based permission model with public key cryptography. This solution as such, mitigates the limitation of a permissionless blockchain and separates securely shared medical data that are stored off-chain. However, FHIRChain does not address patient consent and lacks a scalable architecture that enables authentication management and flexible data sharing through a patient-centred design.

FHIRChain, in their limitations section, states that in future work they intend to deploy their dApp on a permissioned consortium blockchain with trusted parties such as major hospitals and insurance companies. However, in applications where patients perceive insurance companies as a risk due to fear of discrimination resulting from access to their medical records or brain imaging data, a permissioned chain may not be the most appropriate solution.[26].

MedBloc a recently published solution, addresses some limitations of FHIRChain and incorporates a patient-centred design with symmetric cryptography and storing all data on-chain through a permissioned blockchain solution [13]. This solution adds the feature of revoking consent and uses an authentication server, extending from an already published design which leverages off-chain cloud storage to address scalability [8]. MedBloc's design, however, limits itself in lacking a zero-trust model, for access control, with the use of a permissioned chain. A further vulnerability can be argued for their sharing of the patient's symmetric keys (encrypted) along with records to GPs and clinicians, something that would expose a key used to decrypt all patient data, coupled with the scenario of an adversary having access to the private blockchain, which could prove to be a risk. Finally, storing all data on-chain limits the type of data that can be stored and the computational load needed, which could severely affect scalability and most importantly, it would fail to comply with the GDPR.

III. Methodology

To achieve the objective of this project, we evaluated various software development methodologies. While the Waterfall methodology may have been suitable, we decided that the Agile method was the better choice due to its iterative nature. This allowed us to re-evaluate any flawed assumptions early on in the project's development. Additionally, given the privacy sensitivity and security requirements of the project, the Agile approach ensured that the design and implementation of security measures were thoroughly examined throughout the development process.

To address the specialized requirements of blockchain-based development and the sensitivity of privacy, we employed the ABCDE modified Scrum methodology specifically designed for blockchain [20]. To ensure adherence to security requirements and high test coverage, we incorporated Test-Driven Development (TDD) and Behavior-Driven Development (BDD) into the testing process of the blockchain component. The ABCDE methodology differs from traditional Scrum in the following ways:

- The development activities are separated into two flows, one dedicated to the smart contract development and the second to the dApp front-end development.
- A clear definition of the activities that must be performed to design, develop, test and integrate the dApp system with the smart contracts.
- Further emphasis on documentation of the smart contracts using UML diagrams and the BDD test suite, to aid development, security assessment, and visualisation.
- Focused activities related to security auditing.

A. Security and performance assessment

In line with the ABCDE methodology, the first step in security management is to adopt a security-first mindset [20]. To mitigate any risks that may arise from our use of the Agile methodology, which prioritizes simplicity, short iterations, and productivity, we have implemented additional activities focused on security. This is in line with the project's overall goal of securely handling and accounting for data, and avoiding data breaches.

A good starting point for security is consulting well recognised foundations in information security. The open web

This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008,2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

application security project (OWASP) provides a top ten list for proactive controls [2]. These controls will be used as part of the methodology to formulate appropriate security requirements, and to create a secure architecture for the dApp as a whole.

To further enhance the security of the dApp, in addition to following OWASP guidelines, we have also considered and implemented ConsenSys' guidelines for smart contract security [5]. Solidity is the programming language used for smart contract development on Ethereum, and these guidelines provide a comprehensive understanding of security considerations specific to this language. They are maintained and updated by ConsenSys, and also take into account contributions from the broader Ethereum community.

As part of the ABCDE development process, security checklists are utilized throughout the development process, not limited to any specific design, coding, or testing phases. This procedure is particularly important in meeting the specifications for blockchain development and covering known practices to fortify the application against known vulnerabilities, and to enhance data privacy. Relevant patterns were retrieved from the pattern collection for blockchain-based applications [32] to further support the security and privacy of the application.

Additionally, as a recommended security tool by ConsenSys and the broader Ethereum community, we used the Slither framework for automated vulnerability analysis. Slither, which is written in Python 3, runs a suite of vulnerability detectors and enables developers to detect vulnerabilities, improve their understanding of the code, and quickly create custom analyses [6].

B. Gas optimization

Gas is a crucial aspect of blockchain development as it describes the cost of deploying and running smart contracts on the Ethereum blockchain. It was designed to prevent overuse of the blockchain's resources, and is measured in Ether, which can be purchased using traditional currency. To improve the performance of our solution and reduce the cost of deploying and running the smart contracts, optimization is necessary. This is achieved by implementing recognized and tested design patterns for gas optimization, as outlined by Marchesi et al.[18].

The main focus of gas optimization is to further enhance security by preventing DoS attacks and avoiding unwanted smart contract reverts/failures due to running out of gas. Implementing design patterns for gas optimization can also help to reduce the complexity of the smart contracts. In cases where the smart contract system has a limited set of functionality, simplicity can be more effective than complexity [5]. This can help to reduce the attack surface of the application and enhance security, which is in line with the goal of this project. A gas report was used to evaluate the effectiveness of the gas optimization.

IV. Design

With a focus on patient-centered care and taking into account recommendations from the Topol review [29], user-centered design (UCD) techniques such as personas and user journey mapping were employed in this project. These were used to provide justification and visualization of the scenarios in which patients would use the dApp, and also served as a guide for properly defining the requirements discussed further below. Emotion regulation has been identified as a key focus in addressing psychological disorders and enhancing well-being. Some research suggests that low levels of self-awareness are a risk factor for anxiety, stress, and depression [9]. Mood monitoring has been shown to increase emotional self-awareness and in some cases, it has led to the recognition of dysfunctional patterns that can be interrupted through modifications of routines [21]. To enhance the use case of this project and match our main actor, the Perceived Stress Scale (PSS), a classic stress assessment instrument, was used as a concept for patient's personal data. This data would be monitored and logged by the user on a regular basis.

A. Actors and Goals

Following the guidelines from ABCDE, the next step in the design process was to define the system's goal and the possible actors interacting with the system.

The goals of the system are:

- To operate a smart contract access control list that is managed by patients/users.
- To securely store encrypted data pointers for personal mood monitoring.
- To allow patients/users to share their mood monitoring data with therapists if needed.

Actors:

- Patient: Creates medical data from mood monitoring questionnaires on the app system. Manages control and flow of data.
- Therapist: Treats patient and wants to access reports to support treatment.

B. Security requirements and practices

Following the OWASP and ConsenSys guidelines mentioned above a set of security practices is also formed. This forms the basis upon which the security requirements were built and the essential design patterns implemented. The most relevant ones, relating to SCs development were identified according to the security checklists paper [19] and presented below.

- C1: Define security Requirements: To re-enforce the focus on security and meet the aim of this project, accounting for limitations of scrum, security requirements are defined to make sure that the application adheres to recognised principles.
- C2: Leverage Security Frameworks and Libraries: Not re-inventing the wheel is a common practice in computer science. Re-using security hardened software from trusted sources will make the app more secure.
- C5: Validate All Inputs: Validating inputs accounts for data integrity and verification. This will add to the confidentiality and integrity of the application.

This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008,2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

- C6: Implement Digital Identity: Using digital identities from recognised practices will allow limiting unauthorized access and is a core requirement for the access control feature of this dApp.
- C7: Enforce Access Controls: This goes hand-in-hand with the main feature of this dApp as mentioned above.
- C8: Protect Data Everywhere: Being aware that data stored in a SC are always accessible to read, independently of their visibility. Features are needed to account for patient data security.
- C10: Handle All Errors and Exceptions: Even small mistakes in error handling may lead to devastating failures and make the application vulnerable. This is of increased importance for smart contracts.

Design patterns for Security:

1) **Authorization:** *Restrict the execution of critical methods to specific users.*
   Context: Public functions can be called by anyone with a wallet on Ethereum. This would expose data that is stored on-chain.
   Design choice: Embedded addresses to grant permissions pattern [19] - Critical methods should be invoked only by a specific set of addresses, which belong to privileged users/contracts.
2) **Privacy:** *Ensure data integrity, confidentiality and adhere to the GDPR.*
   Context: Metadata stored on-chain are publicly available on Ethereum, this exposes confidentiality.
   Design choice: Encrypt on-chain metadata pattern [32] - By encrypting on-chain metadata, patient data privacy is protected. Both symmetric and asymmetric encryption is to be used to differentiate private records from those intended to be shared.

Design patterns for Gas optimization:

1) Storage patterns
- *Limit Storage:*
  Context: Storage is by far the most expensive kind of memory, so its usage should be minimized.
  Design choice: Limit data stored in the blockchain, always use memory for non-permanent data. Also, limit changes in storage: when executing functions, save the intermediate results in memory or stack and update the storage only at the end of all computations.

2) Saving space
- *Mapping Vs Array:*
  Context: Solidity provides only two data types to represents list of data: arrays and maps. Mappings are cheaper, while arrays are packable and iterable.
  Design choice: In order to save gas, it is recommended to use mappings to manage lists of data, unless there is a need to iterate, or it is possible to pack data types. This is useful both for Storage and Memory. You can manage an ordered list with a mapping using an integer index as a key.

3) Miscellaneous
- *Optimizer*
  Context: Optimizing Solidity code to save gas in an exhaustive way is difficult.
  Design choice: Always turn on the Solidity Optimizer. It is an option of all Solidity compilers, which performs all the optimizations that can be made by the compiler. However, it does not substitute the usage of the presented patterns, most of which need information that is not available to the compiler [18].

Further patterns were used for security and gas optimization which, for the sake of brevity, have not been included here.

C. Architecture

The architecture of the full-stack was changed and revamped throughout sprints.

- AWS was chosen to host the database needed for off-chain data, along with the authorization and encryption features, due to the robust and extensive data protection services available and the experience of the authors around AWS.
- AWS's key management service (KMS) was a recommendation from OWASP, matching the model of 'C8: Protect data everywhere', for both data at rest and data in transit protection, which AWS and DynamoDB support [2; 3].
- DynamoDB with NoSQL matched the pointer system for a simple key-value type of database. Furthermore, NoSQL and DynamoDB aid scalability by supporting a wide variety of object types for storage (besides text), to accommodate future upgrades of the system.
- React.js and ethers.js are widely used frameworks in the web3 space. The authors of this paper had experience using these frameworks, proving better for operation and auditing.

To maintain modularity and upgradability and aid testing throughout sprints and future work, a modification of the MVC design pattern for blockchain was adopted [33]. The components of which are as follows:

Model:

- AWS cloud DynamoDB NoSQL database.
- Ethereum blockchain component holding metadata.

View:

- React.js with ethers.js for the front-end client.

Controllers:

- AWS Lambda-based serverless APIs for record data retrieval and digital identity authorization using AWS Cognito and STS.
- Metamask wallet acting as a bridge between the client and the blockchain component.

Controller-invoked blockchain data connector service:

- Controls the essential encrypted metadata, including the pointers for the database.
- Verifies the integrity of the data stored off-chain by the use of cryptographic hashes.
- Restricts permissions through transparent access control.

This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008,2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

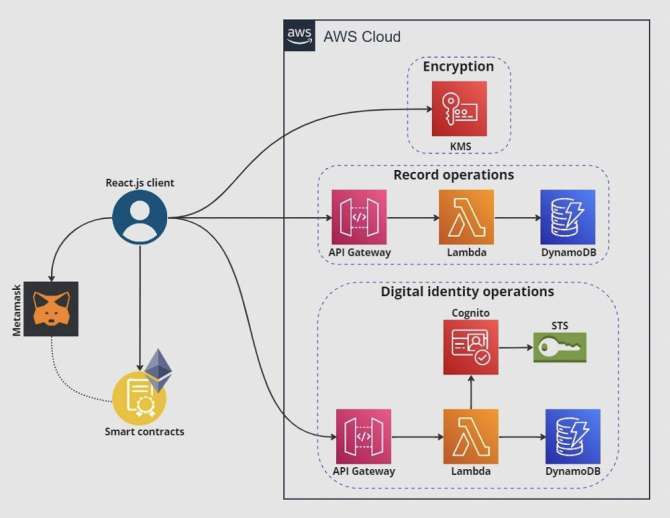

Figure 1: Architecture overview diagram

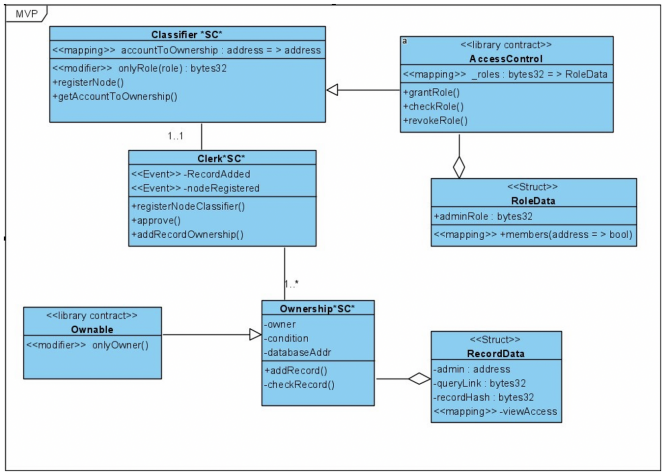

Figure 2: Blockchain component's class diagram

The architecture of the system is depicted in Figure 1, where different controllers with associated services are represented in boxes according to their functionality. To comply with OWASP practices and facilitate the operation of the blockchain component, the Metamask wallet extension was utilized. This component acts as a controller and authenticator, holding the PKI asymmetric keys of the user. Metamask is also linked to an account with a wallet address that can be used to execute transactions on Ethereum.

The dApp client requires the user to connect to Metamask for any functionality to be executed and abides by the authorization pattern. Additionally, the user is registered on the blockchain component with a specified role linking to the wallet address. To further enhance the authorization and security, when interacting with the AWS cloud server, the user is registered with credentials on the AWS Cognito cloud service.

For the use of the server functionality, a sign-verification process occurs on the server, enhancing the security between the components. This authentication flow ties the anonymized wallet address to the digital identity, linking it to the OWASP practice C6 above. This allows for increased security and ensures that the correct user has access to AWS Services, including individualized encryption keys and record data.

D. Encryption

Encryption was a key aspect of this project due to the importance and sensitivity of data handling. To align with the design patterns mentioned above, data both on and off-chain had to be encrypted. To further enhance security, communications to and from DynamoDB use the HTTPS protocol, which protects network traffic by using Secure Sockets Layer (SSL)/Transport Layer Security (TLS) encryption. Additionally, data at rest in DynamoDB tables is also encrypted [3]. An encryption plan was developed as part of the design process to serve as a guideline and complement the security requirements.

- Use of symmetric key (256-bit AES-GCM) server-side encryption of the metadata that will be stored on-chain. Unique keys are created for each patient and limit access to them.
- PKI for sign-then-encrypt for secure data sharing (see Figure 4) [13; 33]
- Keccak256 for record hashing to be stored on-chain.
- Encryption to happen server-side [2].
- Use UUID for a unique and secure id of records, UUIDv4 due to security operation-related context [25].

E. UML diagrams

To aid in development, visualization, and adherence to the designed architecture of the system, UML class and sequence diagrams were created.

The class diagram, as shown in Figure 2, illustrates the interactions between the smart contracts. It was used to visualize the smart contract interactions throughout the development process. The "Clerk" class, in the middle, serves as the core component of the embedded addresses and permissions pattern.

The sequence diagram in Figure 3 describes the user registration on-chain. The user in this case is already signed up with AWS and Metamask. The diagram in Figure 4 describes the functionality of the token-based sharing of the data pointers, this relates to the sharing of records with the therapist through asymmetric encryption.

V. Results

The results of this study demonstrate that the development process followed the agile methodology correctly and was able to adapt the design and implementation as needed to better align with the goals of the project and improve security. Data followed the encryption plan, which enabled securely encrypted pointers with symmetric and asymmetric encryption and integrity verification through hash comparison on-chain.

After implementing refactoring to match the design patterns for gas optimization, a significant decrease in gas consumption

This article has been accepted for publication in a future issue of this workshop, but has not been fully edited. Content may change prior to final publication. Citation information: DOI10.1109/WETSEB59161.2023.00008,2023 IEEE/ACM 6th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)

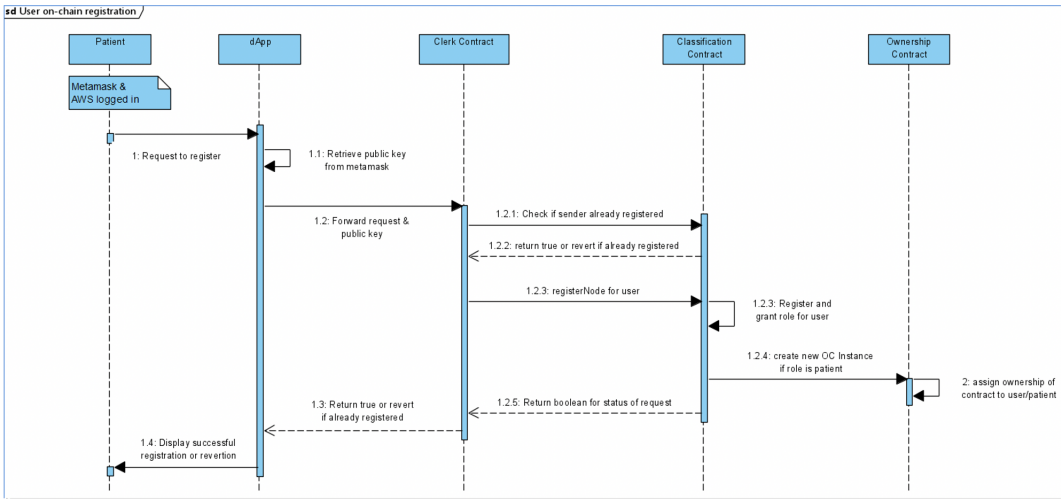

Figure 3: User on-chain registration sequence diagram

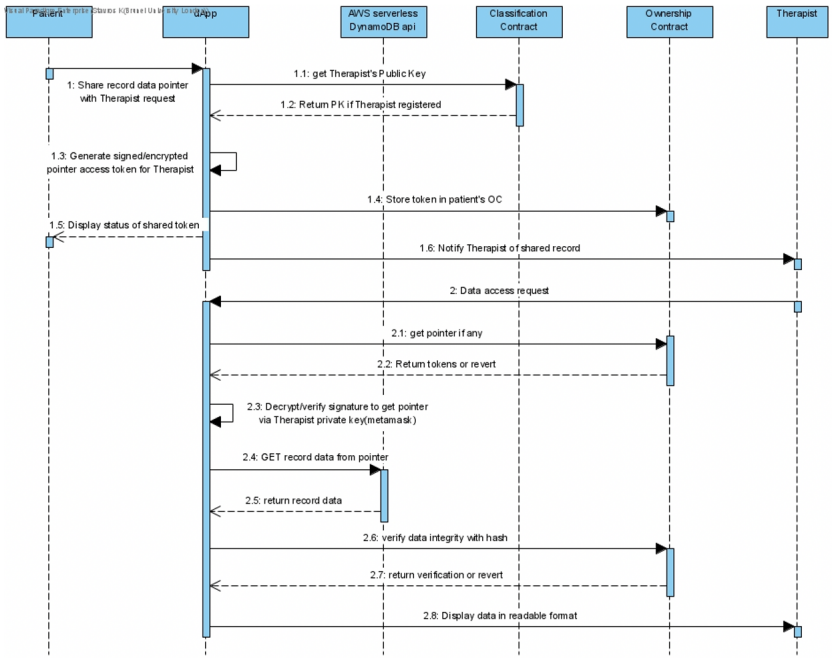

Figure 4: Record data token-based pointer sharing sequence diagram

was observed. The record addition function resulted in a 15.3% decrease, and registration and deployment showed the most substantial reductions at 46.1% and 48.7%, respectively. The gas optimizer was found to be an effective tool for reducing consumption between reports. These findings reinforce the notion that there are significant differences between traditional development and blockchain development, and that these design patterns should be considered a requirement for developing on-chains with a high gas cost.

Furthermore, the utilization of TDD and BDD, along with the functionality and reporting provided by Hardhat, allowed for the achievement of 100% test coverage. BDD was also found to aid in the documentation and conciseness of the test suite. Finally, Slither while not the most vital in weight for assessing security for this dApp, provided useful insights and proposed modifications for a better-optimized application. Therefore, Slither is a useful tool that provides rich detail around the static analysis and should be used alongside manual inspection.

VI. Limitations and Future Works

Further focus on smart contract development and combining the involvement of experts for further manual auditing would increase the security standard as this application scales. However, due to this being a full-stack application, many

features were placed out of scope. Ideas for future work will be discussed here to provide a roadmap for scaling the application and making it more secure and ready for deployment.

Choosing Ethereum as the development chain provided transparency of the access control, but it comes with a non-negligible cost to operate. Changing the chain or architecture might have to be considered moving forward. Polygon is a Level 2 Ethereum-based chain that increases transaction speed and reduces cost substantially, but it might not provide the same level of transaction security. Migrating to Polygon would be a worthwhile consideration [27].

A feature that was considered was separating off-chain data. The inclusion of a separate temporary "box" or database for holding only records that are meant to be shared could increase security. This would mean that when a 3rd party wants to access a patient's record, that record would be copied to the temporary box with a timer for deletion after a reasonable amount of time. These shared pointers would only point to the temporary box. If the metadata were compromised, perhaps by asymmetric or symmetric encryption becoming unsafe in the future, only a limited number of data would be exposed if not already deleted. This would also allow for a secure solution regarding the authorization for a release of records to a research patient registry for a fixed period of time [11].

Another good feature relating to gas optimization would be the use of the data contract pattern. This would work by separating the ownership contract's meta-data management logic, which would be the same for all ownership contracts into one contract and several data contracts holding only the metadata. In this way, when the logic needs to be updated (by using a new smart contract), there is no need to migrate old data [18; 32]. This would theoretically substantially reduce storage and therefore gas consumption.

A common design pattern was the proxy pattern, which incorporated a registry that would route calls to upgraded re-deployed versions of any of the contracts, should any bugs arise. The circuit breaker was placed as a mitigation due to limited time and this project serving as a proof-of-concept rather than a commercial-ready product.

An important addition to this project would be to address the issue of data interoperability, which is a well-known problem in digital health, EHRs, and the NHS. By adopting the OpenEHR standard for data format specification, this project could improve its ability to interoperate with other systems.

A recommendation from MedBloc [13] is to include the feature of revoking consent in data sharing. This, in conjunction with the temporary box mentioned earlier, would increase patients' level of control over their data and enhance the dApp's compliance with GDPR's right to erasure [14].

Monetizing access to personal data is another idea that could be considered. However, this approach could be controversial from an ethical standpoint, as it could leave vulnerable people with limited income unable to protect their privacy.

Furthermore, as this project is a mental health application, a well-designed and developed user experience (UX) is crucial. Features such as psychoeducation, gamification, and further emotion tracking and regulation should be considered. Finally, it is essential to involve clinicians and conduct peer-reviewed acceptability studies throughout the development process in order to create a usable and impactful mental health application [17].

VII. CONCLUSION

This paper described the process of creating a proof-of-concept dApp that prioritizes data sensitivity and patient-centered data management design and implementation. The project adhered to recognized security design patterns and practices, including the use of symmetric and asymmetric cryptography to protect and maintain the integrity and confidentiality of the data. The dApp successfully followed guidelines from OWASP and ConsenSys, and mitigated issues related to GDPR compliance. The security of the dApp was strengthened through the implemented architecture and use of appropriate design patterns. The blockchain component was designed and developed with a focus on low complexity and achieved 100% test coverage for the smart contracts. Gas consumption was also successfully reduced by 15-45%. Suggestions for future work include considering migration to a different blockchain network or implementing features such as data interoperability and data monetization. It is important to note that creating a well-designed and user-centered UX with features such as psychoeducation and emotion tracking will also be crucial in creating an impactful and usable mental health application.

REFERENCES

[1] Agaku, I.T., Adisa, A.O., Ayo-Yusuf, O.A., Connolly, G.N.: Concern about security and privacy, and perceived control over collection and use of health information are related to withholding of health information from healthcare providers. Journal of the American Medical Informatics Association **21**(2), 374–378 (2014)

[2] Anton, K., Manico, J., Bird, J.: Owasp proactive controls for developers. Open Web Application Security Project (OWASP) (2018)

[3] AWS: DynamoDB Encryption at Rest (2022), https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/encryption.usagenotes.html

[4] Buterin, V.: Ethereum White Paper: A Next Generation Smart Contract Decentralized Application Platform. Etherum (January), 1–36 (2014), https://github.com/ethereum/wiki/wiki/White-Paper

[5] ConsenSys: Ethereum Smart Contract Best Practices (2022), https://consensys.github.io/smart-contract-best-practices/

[6] Crytic: Slither, the Solidity source analyzer (2022), https://github.com/crytic/slither

[7] Dagher, G.G., Mohler, J., Milojkovic, M., Marella, P.B.: Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable cities and society **39**, 283–297 (2018)

[8] Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., Wang, F.: Secure and trustable electronic medical records sharing using blockchain. In: AMIA annual symposium proceedings. vol. 2017, p. 650. American Medical Informatics Association (2017)

[9] Eisenstadt, M., Liverpool, S., Infanti, E., Ciuvat, R.M., Carlsson, C., et al.: Mobile apps that promote emotion regulation, positive mental health, and well-being in the general population: systematic review and meta-analysis. JMIR mental health **8**(11), e31170 (2021)

[10] Garrido, S., Millington, C., Cheers, D., Boydell, K., Schubert, E., Meade, T., Nguyen, Q.V.: What works and what doesn't work? a systematic review of digital mental health interventions for depression and anxiety in young people. Frontiers in psychiatry **10**, 759 (2019)

[11] Gordon, W.J., Catalini, C.: Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Computational and structural biotechnology journal **16**, 224–230 (2018)

[12] Healthwatch: How do people feel about their data being shared by the NHS? (2018), https://www.healthwatch.co.uk/report/2018-05-17/how-do-people-feel-about-their-data-being-shared-nhs

[13] Huang, J., Qi, Y.W., Asghar, M.R., Meads, A., Tu, Y.C.: Sharing medical data using a blockchain-based secure ehr system for new zealand. IET Blockchain **2**(1), 13–28 (2022)

[14] ICO: Guide to the UK General Data Protection Regulation (UK GDPR) (2022), https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/

[15] Kassab, M., DeFranco, J., Malas, T., Destefanis, G., Neto, V.V.G.: Investigating quality requirements for blockchain-based healthcare systems. In: 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). pp. 52–55. IEEE (2019)

[16] Kassab, M., DeFranco, J., Malas, T., Neto, V.V.G., Destefanis, G.: Blockchain: A panacea for electronic health records? In: 2019 IEEE/ACM 1st International Workshop on Software Engineering for Healthcare (SEH). pp. 21–24. IEEE (2019)

[17] Koulouri, T., Macredie, R.D., Olakitan, D.: Chatbots to support young adults' mental health: An exploratory study of acceptability. ACM Transactions on Interactive Intelligent Systems (TiiS) **12**(2), 1–39 (2022)

[18] Marchesi, L., Marchesi, M., Destefanis, G., Barabino, G., Tigano, D.: Design patterns for gas optimization in ethereum. In: 2020 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE). pp. 9–15. IEEE (2020)

[19] Marchesi, L., Marchesi, M., Pompianu, L., Tonelli, R.: Security checklists for ethereum smart contract development: patterns and best practices. arXiv preprint arXiv:2008.04761 (2020)

[20] Marchesi, L., Marchesi, M., Tonelli, R.: Abcde—agile block chain dapp engineering. Blockchain: Research and Applications **1**(1-2), 100002 (2020)

[21] Morris, M.E., Kathawala, Q., Leen, T.K., Gorenstein, E.E., Guilak, F., DeLeeuw, W., Labhard, M.: Mobile therapy: case study evaluations of a cell phone application for emotional self-awareness. Journal of medical Internet research **12**(2), e1371 (2010)

[22] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review p. 21260 (2008)

[23] NHS: NHS England » Developing patient centred care (2014), https://www.england.nhs.uk/integrated-care-pioneers/resources/patient-care/

[24] Office for National Statistics: Suicides in the UK - Office for National Statistics. Tech. rep. (2019), https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/deaths/bulletins/suicidesintheunitedkingdom/2018registrations

[25] Peabody, B., Davis, K.R.: New UUID Formats. Internet-Draft draft-peabody-dispatch-new-uuid-format-04, Internet Engineering Task Force (jun 2022), https://datatracker.ietf.org/doc/html/draft-peabody-dispatch-new-uuid-format#section-8

[26] Pilozzi, A., Huang, X.: Overcoming alzheimer's disease stigma by leveraging artificial intelligence and blockchain technologies. Brain Sciences **10**(3), 183 (2020)

[27] Polygon: Polygon PoS - Polygon (2023), https://polygon.technology/solutions/polygon-pos

[28] Reform: Blockchain in the NHS - REFORMER THOUGHTS. Tech. rep., Reform, London (dec 2018), https://reform.uk/publications/reformer-thoughts-blockchain-nhs/

[29] Topol: The Topol Review: Preparing the healthcare workforce to deliver the digital future. An independent report on behalf of the Secretary of State for Health and Social Care. NHS (February), 102 (feb 2019), https://topol.hee.nhs.uk/the-topol-review/https://topol.hee.nhs.uk/wp-content/uploads/HEE-Topol-Review-2019.pdf

[30] Vazirani, A.A., O'Donoghue, O., Brindley, D., Meinert, E.: Blockchain vehicles for efficient medical record management. NPJ digital medicine **3**(1), 1–5 (2020)

[31] Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper **151**(2014), 1–32 (2014)

[32] Xu, X., Pautasso, C., Zhu, L., Lu, Q., Weber, I.: A pattern collection for blockchain-based applications. In: Proceedings of the 23rd European Conference on Pattern Languages of Programs. pp. 1–20 (2018)

[33] Zhang, P., White, J., Schmidt, D.C., Lenz, G., Rosenbloom, S.T.: Fhirchain: applying blockchain to securely and scalably share clinical data. Computational and structural biotechnology journal **16**, 267–278 (2018)