S	
E1	
E2	
For Office Use Only	

Masters Project Final Report

(MCS)

2019

Project Title	Ensuring Data Integrity and Immutability of Audit History Critical System using Blockchains
Student Name	R.M.S. Chathuranga
Registration No. & Index No.	Registration Number: 2017/MCS/012 Index Number: 17440127
Supervisor's Name	Dr. Kasun De Zoysa

For Office Use ONLY

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: R.M.S. Chathuranga

Registration Number: 2017/MCS/012

Index Number: 17440127

Signature: Date: 14/11/2020

This is to certify that this thesis is based on the work of Mr. R.M.S. Chathuranga under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr. Kasun De Zoysa

Signature: Date:

Ensuring Data Integrity and Immutability of Audit History Critical System using Blockchains

A dissertation submitted for the Degree of Master of Computer Science

R.M.S. Chathuranga
University of Colombo School of Computing
2020

Acknowledgement

I would like to express my sincere gratitude to Mr. Sujay Rajanna (resided in Cary, North Carolina, USA) for that exquisite inspiration and motivation injected into my mind to observe into the world of Blockchains. Honestly, his inspiration guided me to research on Blockchains even before the start of this Masters degree program.

I would also like to convey my utmost gratitude to my Supervisor Dr. Kasun De Zoysa, senior lecturer of University of Colombo School of Computing for the valuable guidance and support provided throughout to complete this research.

And I would also like to express my gratitude to my co-supervisor Dr. Miyuru Dayarathna, Senior Technical Lead at WSO2.Inc for the continuous support, guidance and valuable knowledge shared to complete this research.

Abstract

Even though the information technology and computer science has been advanced through decades and accomplished greatest achievements, assuring information/data immutability and integrity profoundly is yet an unsolved challenge. This challenge is highly important for audit history-critical systems in the real world. Anyway, it is not possible to overcome this challenge precisely using the existing highly advanced approaches, technologies, frameworks, protocols related to the domain of data/system security. Information/data can be changed/tampered at different layers and different access points of a system and even if the information is tampered, it goes undetected, which is another critical factor regarding this concern. Therefore, no technology or system claims that their information is 100% tamper-resistant; anyway, the realistic truth is that the aforementioned feature can only be improved, but not profoundly achieved.

This research has used blockchain technology to achieve this feature while strengthening the security of an overall system on system authentication and authorization layer too, in order to achieve immutability. The basic concept adopted is that if the requirement is to make information immutable, information storage should be made tamper-resistant meanwhile securing the approach of how the information is accessed and mutated.

Therefore, in this study, attention is paid to further secure the system authentication and authorization layers too using the blockchain technology. User authentication (login) is implemented with challenge-response protocol with the participation of blockchain so that the user's public key also is stored in the blockchain. Therefore, as long as the information immutability is assured by the blockchain, challenge-response protocol based user authentication will also be highly protected, hence changing keys won’t make it possible for illegal login attempts. User authorization has exploited the concept of claim-based authorization and due to the fact that user ‘role’ claim is protected with the blockchain, authorization process too has been made highly secured.

The proposed solution has been developed with the recently popularized new programming language, Ballerina, for server-side developments and the cutting edge javascript library React, for client-side developments. The prototype application developed is an online news portal where information immutability and integrity is a high concern.

The evaluation of the proposed solution was done to ensure that the expected functionalities of the system can be done without any issues and also by demonstrating security breach efforts to tamper news article data. The solution is finally compared with the existing solutions to highlight the significance of the suggested approach, analyzing the possible drawbacks too. The conclusion of the study is that it is highly effective in using the suggested solution to ensure information immutability and integrity.

Contents

1. Introduction **1**

1.1 Introduction 1

1.2 Problem Domain 1

1.3 Problem 2

1.2 Background and motivation 3

1.3 Exact Computer Science Problem 5

1.4 Research Contribution 5

1.4.1 Aim 5

1.4.2 Objectives 5

1.5 Solution 6

1.6. Scope 6

1.7. Evaluation 6

1.8 Structure of the Thesis 7

2. Literature Review **9**

3. Technology Adapted **13**

3.1. Introduction 13

3.2. Blockchains 13

3.3. Challenge-Response protocol 16

3.4. Claim-Based Authorization 18

4. Approach **19**

4.1 Introduction 19

4.2 Users 19

4.3 Inputs 20

4.4 Outputs 21

4.5 Process 21

4.6 Adoption of Technologies 21

4.6.1 Adoption of Ethereum blockchain 21

4.6.2 Adoption of Challenge-Response protocol 23

4.6.3 Adoption of Claim-Based Authorization 23

5. Design **24**

5.1. Introduction 24

5.2 Abstract Design of the Basic Concept 25

5.3 Top Level Architecture of the Prototype Application 27

5.3.1 Web Portal 28

5.3.2 Business Logic Layer 28

5.3.3 Database (RDBMS) 28

5.3.4 Blockchain 29

5.4 Authentication Model 30

5.3 Authorization Model 31

6. Implementation **33**

6.1. Introduction 33

6.2 System Architecture and Implementation 33

6.2.1 Web Portal 34

6.2.1.1 Introduction 34

6.2.1.2 Single page application 35

6.2.1.3 Component routing mechanism 36

6.2.1.4 How the web app has been served as a service 37

6.2.1.4 Material-UI as UI library 37

6.2.2 The business logic layer 37

6.2.2.1 REST API exposing services 38

6.2.2.2 Implementation of Ballerina Services 39

6.2.3 Database implementation 39

6.2.4 Blockchain Setup 41

6.3 Implementation of News article writing-reading model 42

6.4 Implementation of User information Storage Model 46

6.5 Implementation of User Authentication Model 48

6.6 Implementation of User Authorization Model 51

7. Evaluation **53**

7.1 Introduction 53

7.1 Evaluation Approach 53

7.2.1. Evaluation of the functionality 54

7.2.2. Demonstration of security breach attempts 54

7.2.3. Logical Analysis with State transition/ flow diagrams 56

7.2.4. Comparison with existing solutions 57

8. Conclusion & Further work **60**

8.1 Overall Achievement 60

8.2 Limitations of the Solution Presented 60

8.3 Achievement of the objectives 61

8.2 Further Work 62

References **63**

Appendix A **67**

List of Figures

Page

Figure 3.1 - Basic architecture of a blockchain system 15

Figure 3.2 - Basic flow of processes in a blockchain 15

Figure 3.3 - Data exchange flow in zero-knowledge protocol 17

Figure 5.1 - Abstract Design of the System 26

Figure 5.2 - Basic architecture of prototype application 27

Figure 5.3 - Interconnection between RDBMS and Blockchain 30

Figure 5.4 - Abstract design of the system authentication model 31

Figure 6.1 - index.js and index.html code implementation 35

Figure 6.2 - Web application component/view routing mechanism 36

Figure 6.3 - Data considered to store in Blockchain 40

Figure 6.4 - Architecture of News Article writing and reading model 43

Figure 6.5 - Article validity response 46

Figure 6.6 - Implementation of User info Storage Model 48

Figure 6.7 - Sequence Diagram : Implementation of User Authentication Model 49

List of Tables

page

Table 6.1: REST API Resources 38

Chapter 1

1. Introduction

1.1 Introduction

Maintaining the immutability of the data records is highly important in a system in which the audit trail and history is a critical feature. The original data/information needs to be untampered and even if changed for some requirements, the audit trail and the history of the data has to be recorded. This is significant for the data integrity and credibility of such a system eventually. The 'integrity' stands for maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle.

This problem is vital for applications where reliability, integrity and non-repudiation are key foundations that those systems are based on. E.g. Shared collaborative document/workspace, bank transaction ledgers, audit logs, news portals, etc. These aspects are also significant when two or more parties perform simultaneous transactions or function together, and their all the activities need to be recorded in an incontrovertible manner. Therefore, ensuring the immutability of the data records, audit trail records carry a high value in order to achieve above attributes.

1.2 Problem Domain

The problem discussed above is related to the aspects of the domain of establishing the integrity of information security, system authentication/authorization and addressing them with blockchain concepts.

1.3 Problem

The problem to be addressed related to the aforementioned scenarios and the domain can be described as below. Information/data immutability, integrity are features very hard to achieve via existing highly used approaches and technologies related to the domain of data/system security. There are numerous technologies, frameworks, protocols invented to achieve this feature. i.e. data validation mechanisms, data replication and backup, maintaining audit trails, using immutable databases, access control mechanisms, etc. Even authorized (but malicious) users can sometimes change the records, hence it is difficult to guarantee the immutability of the records. Data can be changed even at very lower levels of information systems such as data transfer, database layer, data storage layer, etc. via numerous security vulnerabilities and using numerous and ever-renewing hacking approaches. Therefore, assuring safety from such these frauds, attacks is not 100% possible via vastly exploited current technologies.

When considering policies for information security within an organization, the CIA triad; standing for Confidentiality, Integrity and Availability is a model designed to guarantee the security in any kind of secured system. Among these three principles, “integrity” is different from the confidentiality and availability of data. It is a key highlight since once the integrity is compromised, there is no way to restore the original data. Hence the data is lost forever. There is no value of integrity compromised data further. The problem discussed here is tightly coupled with this principle of “integrity”.

There are a number of cyber attacks made to the data stores, information systems world-wide over the history, which compromised the data integrity. One most recent and highlighted example is that in 2015, Kaspersky Lab found out a massive cyber-attack which infringed over 100 financial institutes across the globe that siphoned money from account balances for an estimated value of around $1 billion USD [1].

Following is a list of data high-impactful data integrity breaches reported worldwide [2]

- 2008 – Brazilian government systems were infiltrated by hackers and they inflated logging quotas to disrupt logging industry
- 2010 – Stuxnet Worm was used by hackers to make minor changes in Iran's nuclear power program in an attempt to demolish it
- 2013 – The Associated Press' Twitter account was hacked by a Syrian group and they tweeted that President Obama had been injured in explosions at the White House. This single tweet caused a 147-point drop in the Dow.
- 2015 – Investment banking company 'JP Morgan Chase' was breached with attempts at market manipulation
- 2015 – Anonymous released financial reports exposing firms in the US and China trying to cheat the stock market. In one case, damaging the brand reputation of REXLot Holdings, a games developer, which had inflated its revenue.
- 2016 – Both the Democratic National Committee and World Anti-Doping Agency are breached with hackers manipulating their data to embarrass the organizations

These incidents prove the high impact of data integrity breaches and the fact that it is unavoidable irrespective of being high profile organizations, using advanced and latest technologies.

This is a critical aspect to the domain of news/media, i.e. news portal, where news is published and readers access it to read news. This real-world application is selected to demonstrate a solution to this problem discussed. Therefore, this study is to come up with a model which improves the immutability and integrity of news in an online news portal.

1.2 Background and motivation

The integrity/trustworthiness of news is a compelling problem today. Various individuals/groups/parties publish news, as they wish into various forms of Internet media including Social Media, Web sites, Forums, E-news papers, etc. However, it is a generally known fact that a considerable amount of News is incorrect/invalid/fake/tampered. Therefore, when a particular authorized person adds some news content into a particular portal/system, it is possible that some other user or external party could tamper that entry due to various purposes, such as political benefits/interests, financial benefits, etc. One such example is that once some hackers selling access to tamper U.S. news sites [3]. They were offering to sell access to the content management systems of bundles of news sites, which would enable them to edit or add new content. And some others also offered access to a centralized administration panel for a variety of news sites which were largely Southeast Asian, with Saudi and U.S. sites mixed in [3]. Therefore, it is important if there is a mechanism to ensure the integrity/immutability of news articles in the World Wide Web.

The digital news media is getting popular rapidly over the other news media and hence, the security attacks on these digital news media are getting increased steadily. In earlier days, if someone disliked what was reported in a newspaper, they could write a letter to the editor or run a full-page ad combating the piece. But today in the digital media world, what happens is that hacking the particular digital news site/article and changing them or publishing private information/disgusting fake news about them or shutting down their websites completely. Therefore, the security and integrity concerns regarding the digital news domain are highly important today [4].

Common attacks on news sites are Distributed Denial-of-Service (DDOS) attacks but that doesn't affect the integrity of the news content. It directly matters for the availability of the content. But the next major type of attack, called 'Doxing' where attackers obtain and publish personal information about those journalists, politicians, etc. with the objective of maliciously exposing their vulnerabilities. This has become more frequent in the past few years. This is a high example where the data immutability is breached and hence the integrity is compromised.

Considering the above factors, it is highly important to improve the digital media systems, so that they couldn't be vulnerable to data integrity breaches.

1.3 Exact Computer Science Problem

The aim of this research is to find a solution for the aforementioned real-world problem, via the application of computer science. Therefore, the exact computer science problem, to be addressed through this research can be stated as the difficulty of ensuring information/data immutability and integrity in systems.

Therefore, this research will find a resolution for this exact computer science problem and resolve the real-world problem described in previous sections.

1.4 Research Contribution

1.4.1 Aim

The aim is to develop a system which can increase information/data immutability and integrity of audit trail critical systems.

1.4.2 Objectives

- Designing a solution to overcome the problem of assuring immutability and integrity of audit trail critical systems using blockchain and related technologies after a critical study of the problem and literature.
- Designing a solution to improve the security of user authentication process of a system, concerning on preserving the integrity of information used for the authentication process.
- Designing a solution to improve the security of user authorization process of a system, concerning on preserving the integrity of information used for authorization process.

- Implementing a prototype application based on the designed solution approach, using the appropriate solution protocols, technologies and concepts. This prototype application will demonstrate the solution proposed for the original problem.

1.5 Solution

The solution to the discussed problem is to discover, design and implement a model using the blockchain concepts and related technologies, that will ensure the immutability and integrity of news articles stored in an online news portal.

1.6. Scope

Study and come up with a solution to ensure the immutability of data/records of a selected audit trail critical application; online news portal, using blockchain technology. Therefore, the solution will be provided considering the use cases of this application domain, as a prototype application to demonstrate the solution. Management of the history or allowing updates for the data is not covered in this research.

1.7. Evaluation

Since this research is mainly expecting to come up with a protocol, architectural model to a real-world application domain, evaluation of the project will be done based on a simple proto-type application developed related to the selected application domain. The selected application domain is an online news portal. A list of real-world use cases related to the application is considered and proved that the system functions as expected, with regards to the aspects of maintaining the immutability of data/records, information reliability, creditability, non-repudiation and the general authentication, authorization. Furthermore, demonstrations of attempts were done to breach the security of the system and assured that the above aspects are maintained.

Further, the system was logically analysed with State transition/ flow diagrams to show that the system flow won't be locked at any state and would function as expected. Finally, the features of the system were compared with the other literature, highlighting the benefits and advantages of this solution proposed in this study.

1.8 Structure of the Thesis

Chapter 1 has provided the introduction to this thesis describing the problem addressed and the problem domain, background and motivation, research contribution with the aim and objectives of this research, proposed solution, scope and the evaluation approach in brief.

The second Chapter will broadly review on the others' work related to the domain of information security addressing the integrity aspect of information security mainly discussing on the efforts made with the use of Blockchain technology.

Chapter 3, discusses the basic technologies used in this study namely, the blockchain technology, challenge-response protocol and claim based authorization. Chapter 4 describes the approach followed in this research by depicting how the previously mentioned adopted technologies were exploited to solve the said problem, with reference to users, inputs, outputs, process, technology that implements the solution.

Chapter 5 illustrates the design of the solution implementation describing the conceptual and physical modules of the system, also stating how the authentication and authorization model is implemented. Next the Chapter 6 -'Implementation', provides an in-detail description on how the solution is implemented on top of the previously described design., with description on how each module is implemented.

The 7th Chapter; Evaluation provides information on how the implemented system is evaluated based on the functionality, evaluation using demonstrations of security breach attempts, evaluation using state transition diagrams and comparing with existing other solutions. Finally, the 8th Chapter concludes the overall achievements quantitatively and qualitatively in the first place and after that, the achievement of each objective is described. This chapter also mentions problems encountered, limitations of this solution, and some further work could be done to improve the proposed solution.

Chapter 2

2. Literature Review

The problem statement discussed is a widely studied domain, mainly with the emergence of blockchain technologies, with regards to multiple real-world application domains. But the solutions are not highly implemented and injected into actual real-world production systems. Even when considering the domain of ‘News’/media, the concept is neither much discussed nor done related research in masses.

The NewYork Times is currently doing research using blockchain technology to prove the authenticity of journalistic content under a project called "News Provenance Project" [5]. Their primary efforts involved exploring and investigating on blockchain-based systems for the purpose of recording and sharing metadata about images, media and videos published by news organizations. They are trying to establish a set of signals/metadata that can travel with published media anywhere that material is displayed, and use Hyperledger Fabric framework to maintain that data in a blockchain [6]. Having to carry this metadata with the journalistic content can be considered as an additional overhead in this solution.

WSO2.Inc is studying and doing research related to Evaluating Blockchain for Enterprise Integration. [7] They have suggested a blockchain architecture pattern called “Auditable history or workspace pattern”, where the proposed system record actions and create entries/transactions in the blockchain for each of those records. The entry stored in the blockchain contains the hash of activity records, and consequently, the records can’t be disputed later [8].

B.Dunken *et al.* have done a research on Creating an Immutable Database for Secure Cloud Audit Trail and System Logging [9]. They have evaluated the possibility of using conventional RDBMS software to store Audit Log records, removing the record modify/delete operations. Hence the attackers won't be able to modify the audit log data through the RDBMS operations even though a security breach takes place. But the problem here is that an attacker can still tamper the stored data at a lower level of the storage system. They have also suggested using external Archive databases to store audit trail records.

The concept of immutable databases is studied and researched by multiple efforts and they are currently used in production environments by multiple enterprises. Datomic (developed by Cognitect Inc.) and Mentat (developed by Mozzilla) are such immutable databases. The basic concept behind these is that all the data in the database is never deleted, but updated always, keeping the history of all the records/cells. The RDBMS cells maintain a time handle and whenever the cell is updated, a new entry will go into the cell and that time handle will be pointed to the new entry. This allows updating a cell and but still keep the history to maintain many different versions of the cell; to store the whole history of all the values this cell has ever held. Advantage of this is that the mutation of data is not possible by design, in the DBMS itself. But data is stored in versions and even if some unauthorized user changed a data cell, it will be recorded as a new version of the cell value, hence detecting/monitoring that also would need another mechanism. Another disadvantage is that it will need additional storage space for the historical data [10].

A research has been done using Federated claims-based authentication and access control in the vehicular networks in [9], [11]. In this type of authorization mechanism, controlling the access of the invoked service operations is done according to the claims included in security tokens provided by the service consumers. This authorization model can be coupled with Blockchain technology, where the claims of users can be stored in a blockchain.

Another research [12] has proven how to use Challenge-Response Protocol (also known as Zero-Knowledge Proof) with blockchains to implement an authentication model to an application. It has used blockchains smartly to store hashes of public/private keys in public key (asymmetric) authentication. These approaches can be used to secure the audit trails and auditable workspaces.

In a study and implementation done by Research Center of Cyber Intelligence and Information Security, La Sapienza University of Rome [13], they have used a design with two-layer blockchain in which the first-layer ensures adequate performance and the second layer ensures strong integrity guarantees through a principled exploitation of the second-layer. Both layers maintain individual DBMS replicas for each mining node. The first layer uses a lightweight distributed consensus protocol that assures low latency and high throughput. It also doesn't include a PoW (proof-of-work) process before storing the data, hence provides weak data integrity guarantees. The 2nd layer is designed as a PoW-based blockchain that stores a part of evidence of the database operations logged by the 1st layer. These evidences are stored with robust data integrity guarantees but with poor performance which is a known weakness in blockchains. The pros of this study is that it has addressed the issues of high latency and low throughput in a typical blockchain solution. But the disadvantage of this can be mentioned as the complexity of the design, increased infrastructure and maintenance cost.

Another study has demonstrated how to control the access into a RDBMS through a blockchain-based distributed ledger, where the access to the RDBMSwill be provided based on particular consensus in between all the nodes of the blockchain. It stores authorized database operations as smart contracts. Data will be stored in the distributed ledger when the client application makes a request to update/view or store data values in the cloud server. Then it will be validated against the smart contract of the application. If the data retrieval (access) request is succeeded, a notification is sent to the application and simultaneously, a hash pointer is sent to another blockchain layer to access data. This study was demonstrated for data access protection of an IOT application. The key highlight of this study too is that it uses two separate layers of blockchains and one layer acts as a separate access controlling security layer, hence has a complex design and higher infrastructure requirement. [14]

According to the above described studies, it is apparent that the concept of using blockchains to protect data integrity is yet in the experimental phase and each approach could be used for particular application domain and system-layer. They have their own pros and cons too and the domain is still open for further advancements to ensure data integrity and immutability.

Chapter 3

3. Technology Adapted

3.1. Introduction

This research has used multiple cutting edge technologies to develop a solution to the stated problem. The domain of the project is expanded into system security and data integrity which are highly emphasized concerns in modern information technology. There are numerous technologies prevailing, which can be used to address this research problem; few amongst them have been carefully selected to use.

Therefore, the major technologies used in this research are,

1. Blockchains
2. Challenge-Response protocol
3. Claim-based authentication

Below is a brief description of each technology mentioned above.

3.2. Blockchains

Blockchain can be introduced as a technology for distributed ledger, based on a P2P (peer-to-peer) topology, which facilitates data to be stored globally on thousands or millions of servers while allowing anyone on the network to observe everyone else's entries in near real-time. This nature makes it hard for one user to change the entries in the network or to gain control of the whole network.

The blockchain concept is typically exploited by the crypto-currency applications but also been used (and been researched to be used) in multiple several domains such as healthcare, insurance, advertising, copyright protection, energy trading markets, energy Financing, electric vehicles, blockchain music, blockchain government, intellectual property management systems, digital asset delivery, IoT and many other [15][16.] These domains are utilizing the basic superior features in blockchains such as security, scalability and efficiency.

When we say the words “block” and “chain” in this blockchain context, we are actually talking about digital information (the “block”) stored in a public database (similar to a “chain”). “Blocks” on the blockchain are built with a cluster of digital bits of information. A block has basically three components. One is information about transactions such as the time, date and money amount, etc. Next part is about whom participated in a transaction. The other part is information that distinguishes a block from another (hash).

For the universal problem of data protection, privacy and maintaining reliable information trace, the currently existing conventional solutions are still vulnerable to loss of information, leakage of privacy and other types of attacks. But then the blockchain technology was emerged. [15] Blockchain can record and store historical data by establishing a collectively maintained and tamper-resistant public ledger to ensure the reliability and security of the data stored in a dispersed network. In a blockchain, each information collection (transaction), will be packed into blocks by miner nodes and all the blocks are tightly linked to each other via hash values of each previous block and next block, etc.

This basic concept has been used in this research to store news article data, user information, user public keys in a secured, tamper-resistant, immutable manner.

Figure 3.1 illustrates the basic architecture of a blockchain. It has shown how each node is interconnected as a P2P (peer-to-peer) network and how the blocks are interrelated with the hash values of each previous block.

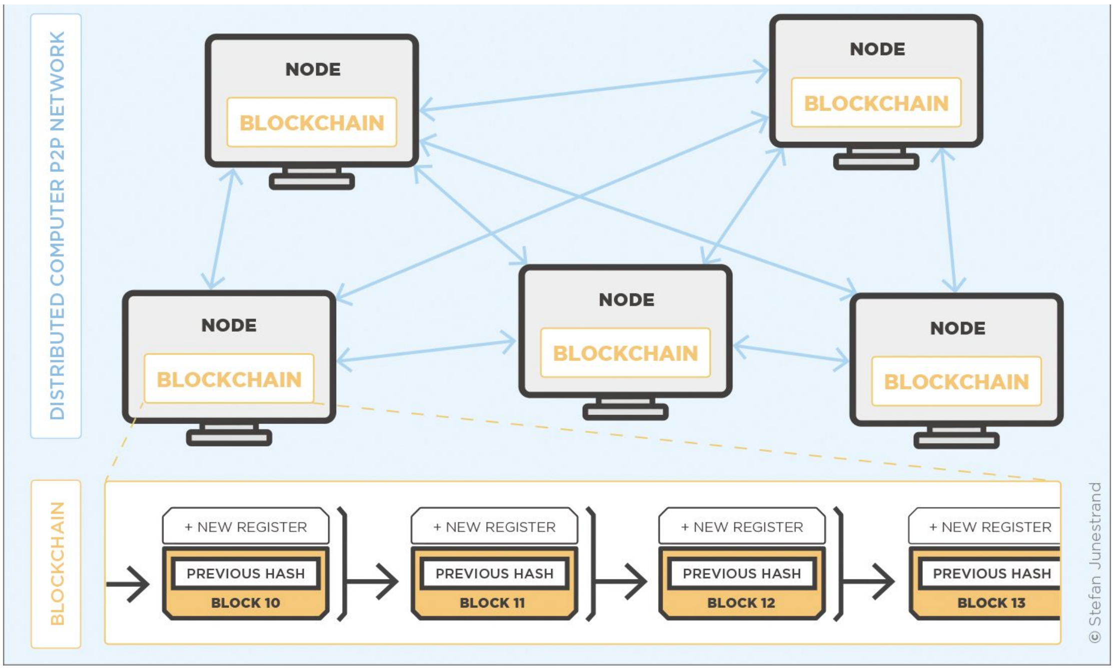
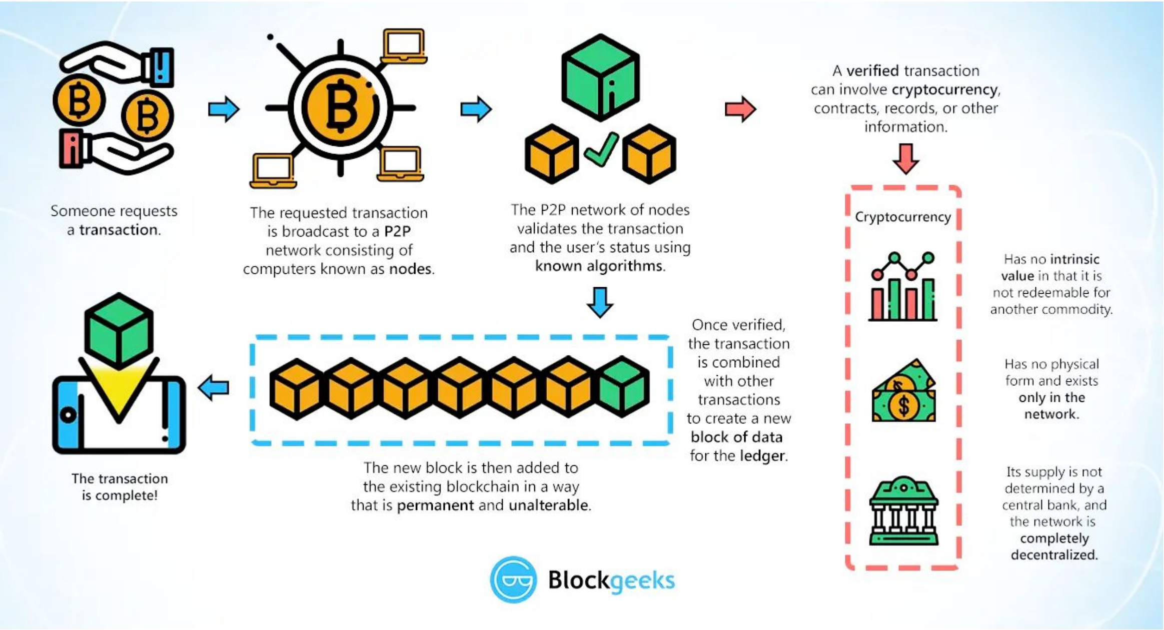


Figure 3.1: Basic architecture of a blockchain system [17]

Figure 3.2 depicts the basic flow of the processes taking place in a blockchain.



Someone requests a **transaction**.

The requested transaction is broadcast to a **P2P** network consisting of computers known as **nodes**.

The P2P network of nodes validates the transaction and the user's status using **known algorithms**.

A **verified** transaction can involve **cryptocurrency**, contracts, records, or other information.

Once verified, the transaction is combined with other transactions to create a new **block of data** for the **ledger**.

The new block is then added to the existing blockchain in a way that is **permanent** and **unalterable**.

The transaction is complete!

Cryptocurrency
Has no **intrinsic value** in that it is not redeemable for another commodity.
Has no physical form and exists **only in the network**.
Its supply is not determined by a central bank, and the network is **completely decentralized**.
Blockgeeks

Figure 3.2 : Basic flow of processes in a blockchain [18]

Blockchains can be used to store any kind of information rather than the typical ‘transaction data’ in which the crypto-currency based applications are using. Therefore, any kind of data records, history records, information can be stored in blockchains and they are stored as an information transaction in the blockchain.

There are multiple types of blockchains popular in the world. Some of them are Ethereum, Cosmos, Cardano, EOS, Hyperledger, Bitcoin, etc. This study has used Ethereum since it is one of the leading cryptocurrencies and blockchains, it’s convenience of usage, rich REST API support, immunity for any third party intervention etc.

3.3. Challenge-Response protocol

Challenge-response protocol or the zero-knowledge-proof is in general encryption schemes used to prove that you know something without revealing what it is. For example, you can show without a doubt that you know the answer to a puzzle without actually disclosing the solution [19].

In the domain of information/system security, this is a process in which a system authorization/authentication is granted by checking the ability to decrypt a random message(challenge) encrypted by the public key of the user (“Prover”/who is going to get the access). Thus, if the user is legitimate he should be able to decrypt the encrypted message and win the challenge, since he is having the private key which is bound to that encrypted public key. Therefore, once the user provides back the correctly decrypted message (correct ‘response’), the resource requested by the user is granted to him by the system (“Verifier”).

Figure 3.3 illustrates the basic data flow of a typical zero-knowledge-proof based application.

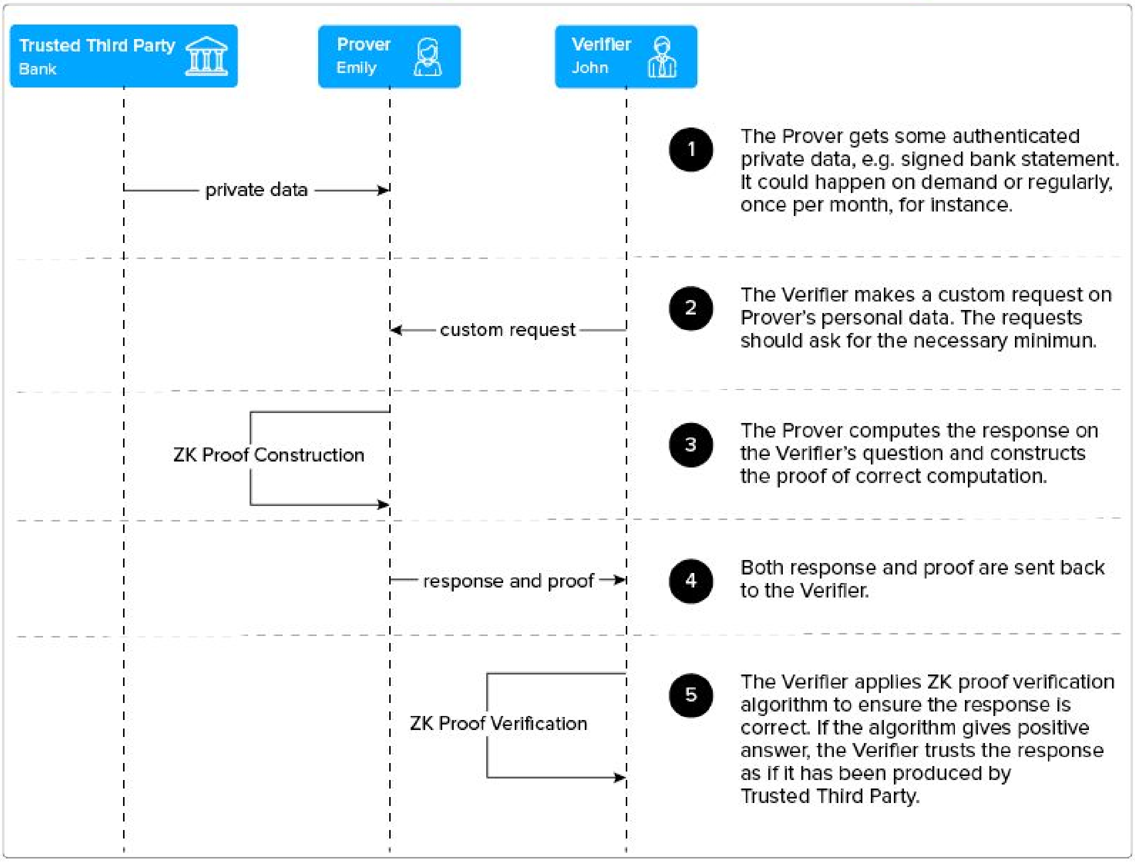

Trusted Third Party
Bank
Prover
Emily
Verifier
John
private data
custom request
ZK Proof Construction
response and proof
ZK Proof Verification
1
The Prover gets some authenticated private data, e.g. signed bank statement. It could happen on demand or regularly, once per month, for instance.
2
The Verifier makes a custom request on Prover's personal data. The requests should ask for the necessary minimun.
3
The Prover computes the response on the Verifier's question and constructs the proof of correct computation.
4
Both response and proof are sent back to the Verifier.
5
The Verifier applies ZK proof verification algorithm to ensure the response is correct. If the algorithm gives positive answer, the Verifier trusts the response as if it has been produced by Trusted Third Party.

Figure 3.3: Data exchange flow in zero-knowledge protocol [20]

Benefits of zero-knowledge-proof can be listed as simplicity (it does not involve any complex encryption method.) and security (It doesn't require anyone to reveal any sort of information.) The disadvantages are that the process is lengthy, imperfect as the messages delivered to verifier/prover might be destroyed or modified, limited since the zero-knowledge protocol demands the secret to be a numerical value (In other cases, a translation is required) [20]. There are two types of Zero-Knowledge Proofs.

1. Interactive Zero-Knowledge Proof -

A prover plays out a progression of activities under the mechanism of mathematical probability to persuade the verifier of the specific fact.

2. Non-Interactive Zero-Knowledge Proof (NIZKP) -

This proof does not need an interactive process. Hence the prover can generate all the challenges at once and leave for the verifier so that verifier can respond to them later.

3.4. Claim-Based Authorization

Claims can be introduced as the user data and they are issued by a trusted source. Claim based authorization simply checks the value of the user claims and allows access to the system resources based on the value of claim(s). If token-based authentication is implemented in a system, a claim may be added within a token by the server (Security Token Service) that generates the token [21]. The claims can be extracted from the token whilst the authentication process via a token introspection operation.

Claims can contain information about the user, roles or permissions, valuable to generate flexible authorization model. Token contains at least one claim and each contains some particular information about the user. The token is digitally signed by the token issuer when it's generated, with the expectation that it will be verified at the receiver's end. A token can likewise contain extra data, for example, expiry date or id too [22]. In a typical Security Token Service (STS) based authentication model, identity provider includes both these STS and an Identity Store/User Store. STS has the responsibility to handle user requests and creating tokens. To verify user identity STS acquires information stored in the Identity Store.

Claim-based authorization/authentication solutions are being increased nowadays. There are many open-source and commercial solutions widely used such as Oracle STS, ADFS (Microsoft), WSO2 Identity Server, Thinktecture IdentityServer, etc.

Chapter 4

4. Approach

4.1 Introduction

The technologies described in chapter 3 are used to develop the prototype system; online news portal in which the data immutability and integrity are increased to a highest level whilst the system authentication and authorization steps are further secured using the same technologies related.

The blockchain technology is used to store and assure the integrity of the news data and the user permissions and roles of the online news portal. Challenge-response protocol is used in-conjunction with blockchains, for a more secure user authentication approach. And a simple implementation of the claim-based authorization concept is utilized to authorize users into the particular resources in the system.

4.2 Users

Users of this system can be categorized in the broad sense as below.

1. News Readers

   The general public who use the online news site to view/read news. Generally, most online news portals provide access to any user to access and read news on their portal without going through any kind of user authentication/authorization process. In this study, this is changed in order to achieve simplicity, to concentrate more on the actual research problem, to be solved via the study. Thus, news readers too need to login to the system to read/view news in the portal.

2. News Writers

   The users working in the particular news agency as actual news writers (editors/reporters). They have the permissions to read the news too.

3. Moderators

   The moderator users are the ones who can assign roles to the users registered into the system. They also have the permissions to do the tasks done by both news readers and news writers.

4. Admin Users

   Admin users have all the permissions, which Readers, Writers and Moderators possess. Other than that, they can also delete users in the system.

There are roles defined for the above-mentioned types of users in the system. Therefore, the role names would be semantical to above user categories above and the permission underneath those roles are as mentioned above. Therefore, the roles defined in the system are as below.

1. Reader
2. Writer
3. Moderator
4. Admin

4.3 Inputs

Inputs of the system can be described respective to each module and use case in the system. But since the outcome of this study is an application which has certain improved security features, these inputs and outputs cannot be presented straightforwardly as inputs and outputs (or results) of the research. Considering the use cases of the system, inputs of the system are,

- User registration: User information, User public key
- User login : user name, user private key
- Add news article: News article information

4.4 Outputs

Outputs of the system can be stated as an online news portal, in which the integrity of the news articles published in the portal is assured to a highest level. An online news portal is developed as a prototype application to demonstrate the proposed solution.

4.5 Process

The prototype application implemented is designed to follow specific processes in order to achieve certain intended attributes in the system. The basic modules of the system are described in detail in the next chapters. The processes adhered are described in brief respectively to each technology used in the system, in the following section. And in detail description of each module and processes are explained in the 'Design' and 'Implementation' chapters.

4.6 Adoption of Technologies

4.6.1 Adoption of Ethereum blockchain

Usage of the Ethereum blockchain in the developed prototype application can be summarized as below.

1. Storing hash values of news articles

Complete news-article content data is not stored in the blockchain, in order to reduce the degraded performance of the blockchain with excessive data [23]. The data is being stored in the Ethereum blockchain with blockchain transactions. There is no maximum block size for Ethereum, while Bitcoin has a fixed block size of 1 MB. Ethereum's block size varies based on how many units of gas can be spent per block.

However, there is a cost for the data storage in any blockchain. The Ethereum blockchain utilized in this application is a 'Private Blockchain'. But in the end, all the data we store in blockchain will be stored in all the nodes in the blockchain too. Therefore, storing all the raw data of the application into the blockchain will cost highly for the storage. When the block size increases, the transaction speed and throughput of the transactions will be reduced too. Therefore, the intuitive option opted is to store only a fixed size hash value of the news article data in the blockchain (within blockchain transactions), rather than the complete raw data. The actual raw data is stored in a separate MySql RDBMS with a mapping to the blockchain transaction data (transaction ID).

2. Storing Hash values of user information

User information is hashed and stored in the Ethereum blockchain as transactions. The user information includes email address, first name, last name, age, public key, roles.

These two types of data are stored in the blockchain to assure immutability and integrity of them. Since the user information too is stored in the blockchain, the user authentication and authorization processes include a blockchain interaction hence, they are protected by the blockchain principles. Tampering user credentials in the RDBMS won't be simply possible to access the system illegitimately, due to this.

4.6.2 Adoption of Challenge-Response protocol

Challenge-Response protocol is used to authenticate users into the system. When user is trying to login to the Web application, the underlying backend service layer provides the challenge (random key encrypted with user's public key) to the user (browser) in return. Web browser decrypts the challenge using the user's private key which is submitted into the browser at the initial login attempt. The decrypted challenge is sent back to the service, where the challenge is validated and it generates a user login session and returns to the browser with login success message, then the user is redirected to the web application home page where he can perform the operations required.

4.6.3 Adoption of Claim-Based Authorization

As stated previously, the claim-based authorization adopted is a simplified process of standard claim-based authorization. It is not the standard claim-based authorization but implemented here to demonstrate the strength of it when used with the blockchain technology.

Therefore, the user claim 'role' is stored in the blockchain along with the user information. (this is stored as a hash as mentioned earlier). Therefore, after the user authentication succeeds, the role claim is extracted from the user information. Based on the role value(s) residing there, the user will be provided access to certain resources in the web application and provided access to the REST API resources which expose the services in the system.

Chapter 5

5. Design

5.1. Introduction

‘Auditable History or Workspace’ is a one mainstream Blockchain Architecture Pattern [8] and the solution for the problem addressed in this research, will be designed and implemented around this concept.

This category of applications provides an auditable history or an auditable shared workspace. Here the blockchain is used as the system which keeps records of activities and generates relevant entries in the blockchain for each of those records. Blockchain is a type of a linked list (e.g. a chain) of blocks which provides shared, immutable, append-only data storage. Blockchain facilitates applications which require tamper-evidence [9] and applications which require transparency, to gain those features. Since blockchains cannot be (conceptually) tampered, the records can't be disputed later. This is the main concept behind this.

Therefore the proposed solution is designed using a blockchain, specifically to store the information which requires it to be immutable. Other than that, the user authentication and authorization objectives too has got the aid of blockchain to improve security.

In this chapter, the abstract design of the basic concept exploited in this research is illustrated initially. Then the top-level physical architecture of the system is described. Then the most important two features which support the proposed solution of this research; system authentication and authorization; are described in abstract.

5.2 Abstract Design of the Basic Concept

The basic concept brought forward in this study is on how to maintain immutability and integrity of news article content in an online news portal. Figure 5.1 illustrates the abstract design of this basic concept within the proposed system. There are two basic types of users in the system namely News writer and News reader.

News writer login to a news writing module of the web portal through a web browser. He will go through authentication and authorization processes. (these steps will be described further in the next sections).

Once the user is checked for his authorization level (permission whether allowed to write news), he will be redirected into the news writing page of the web portal. When the writer writes and saves some news content, a SHA256 hash of the news entry will be generated. This hash is generated for all the text content and other metadata of the news entry. This hash is then base16 encoded and stored into the blockchain. The blockchain transaction hash (record ID) of this news entry is retrieved and it is treated as the news article ID here onwards. This hashing mechanism is implemented to reduce the size of data sent to the blockchain. We use a separate RDBMS to store the actual news entry, against the aforementioned blockchain transaction hash which acts as the news article ID. Hence there is a one-to-one mapping in-between blockchain and actual news data store (RDBMS) for each news entry fed into the system.

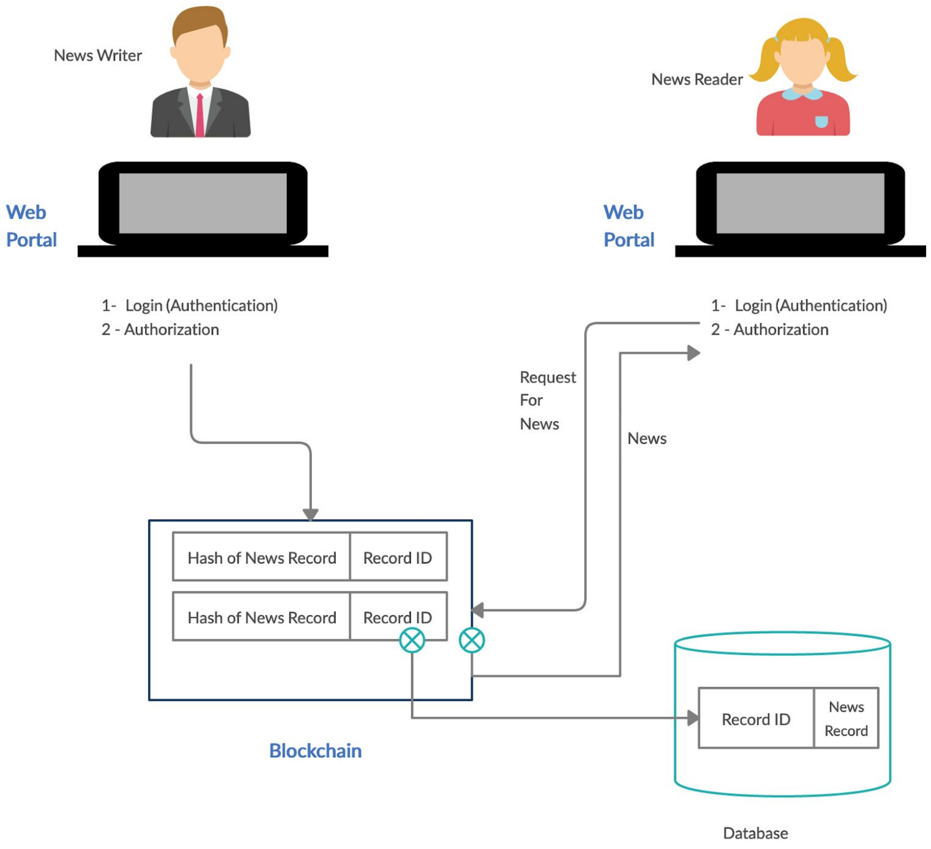


Figure 5.1: Abstract Design of the System

Therefore when the hash of a news entry is stored into the blockchain, it is also recorded in the RDBMS. The entry stored in the RDBMS contains the related blockchain transaction ID (news articleId) and the actual news entry in an appropriate schema format.

Now, the news entry is safely stored in the blockchain and the database. If there is any effort for tampering of the news entry data (hash value) in the blockchain, it is denied by the blockchain; this immutability attribute comes by design with the Blockchain. Even if there is a tampering of data in the database, it is detected since then the hash value of the database entry will not be compatible with the hash value maintained in the blockchain.

When an authenticated, authorized user (news reader) sends a request to the blockchain to get news entries, mapping news entry will be fetched from the database and served back to the reader.

This is the design of the basic concept of News Data securing in the system. How the authentication and authorization processes implemented will be described in the next sections.

5.3 Top Level Architecture of the Prototype Application

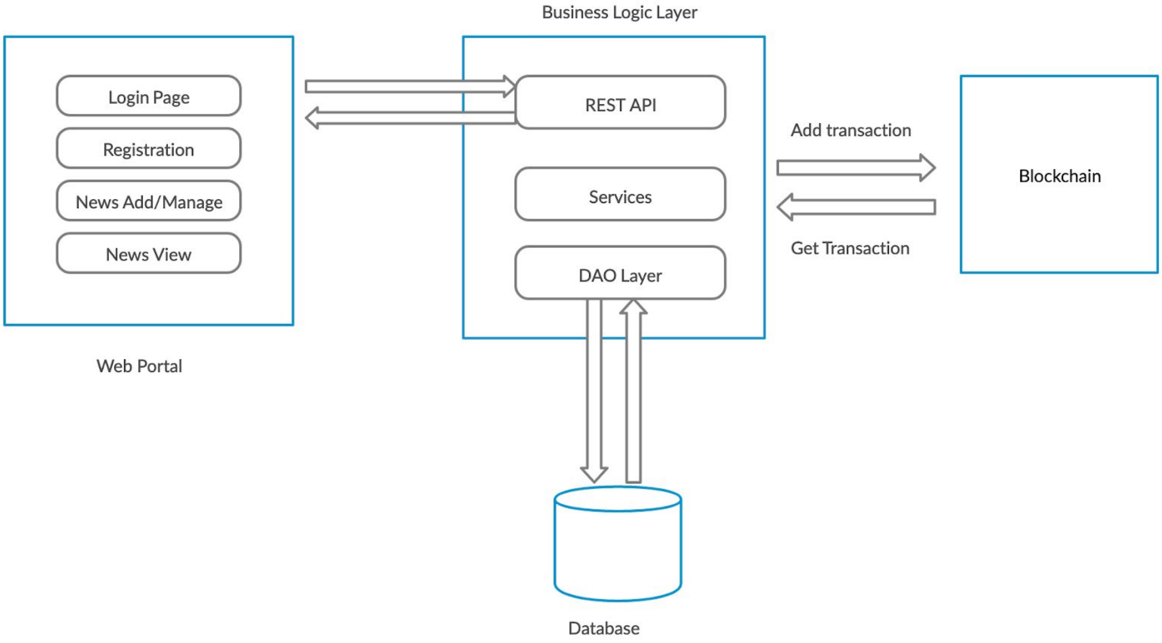


Figure 5.2: Basic architecture of the prototype application

Figure 5.2 illustrates the basic top-level architecture of the prototype application. This depicts the separate physical entities of the system with their inter-connections. According to the diagram below entities can be listed.

1. Web portal
2. Business Logic Layer

3. Database (RDBMS)
4. Blockchain

5.3.1 Web Portal

The web portal is the user interface with which each user interacts with the system. Since this is a News Portal system, the major components in this web portal are user login page, user registration page, news add/manage page and the news viewing page. Each of these web portal sections communicates with the data storage entities (blockchain and the RDBMS database) through a REST API exposed in the business logic layer of the system.

5.3.2 Business Logic Layer

This module consists of the services where the business logic is implemented. Services are exposed as a REST API and the web portal (client) needs to be authenticated and authorized to access these REST API resources. Details on authentication and authorization which use specific protocols are described in later chapters.

The services implemented in this module access the RDBMS database to read/write news article data and user information data through the DAO layer implemented within.

5.3.3 Database (RDBMS)

The database used in the system is a RDBMS. This is used to store news and user information of the system. RDBMS with the blockchain can be introduced as the storage component of the system. RDBMS stores the entire data stack and the blockchain acts as a supporting storage which ensures the immutability/integrity of the data stored in the RDBMS.

The interconnection in-between RDBMS and the Blockchain is illustrated by Figure 5.3. The Web application requests for New Article data from the RDBMS and so it will be queried from the RDBMS. Then the related blockchain transaction is fetched from the blockchain using the blockchain transaction ID which is stored in the RDBMS. These two are compared for the similarity of the hashes and the validity of the news article data is determined before it is sent back to the web application.

In the same way, user information also is stored in the RDBMS and related hash is stored in the blockchain. Integrity of the user information fetched from the RDBMS is validated using the same above mechanism, in the processes of user authentication and authorization.

5.3.4 Blockchain

The blockchain is used to store information with blockchain transactions. The blockchain consists of a number of nodes and the proposal is to establish blockchain nodes per each news writer. Therefore, each news writer will maintain his own copy of data in his blockchain node. Thus assume if there are 3000 news writers across the globe in this organization, there will be 3000 nodes (so 3000 miners) in the blockchain. Therefore, if any intruder intends to change data related to a newspaper article he will need to intrude at least 1501 blockchain nodes according to the theory of ‘51% Attack’. Even he was able to do that, changing all the required historical blocks needs huge processing power and technically it is believed impossible [24], rather than adding new transactions. Hence it is logical to state that it is highly impossible to alter an already published article, but if the attacker succeeds with 51% attack, (which is also very difficult and rare), they can post their fake articles. Anyway even a group of miners managed to add fake articles or update existing articles, they would need to hack into the RDBMS too to reflect them in the news portal.

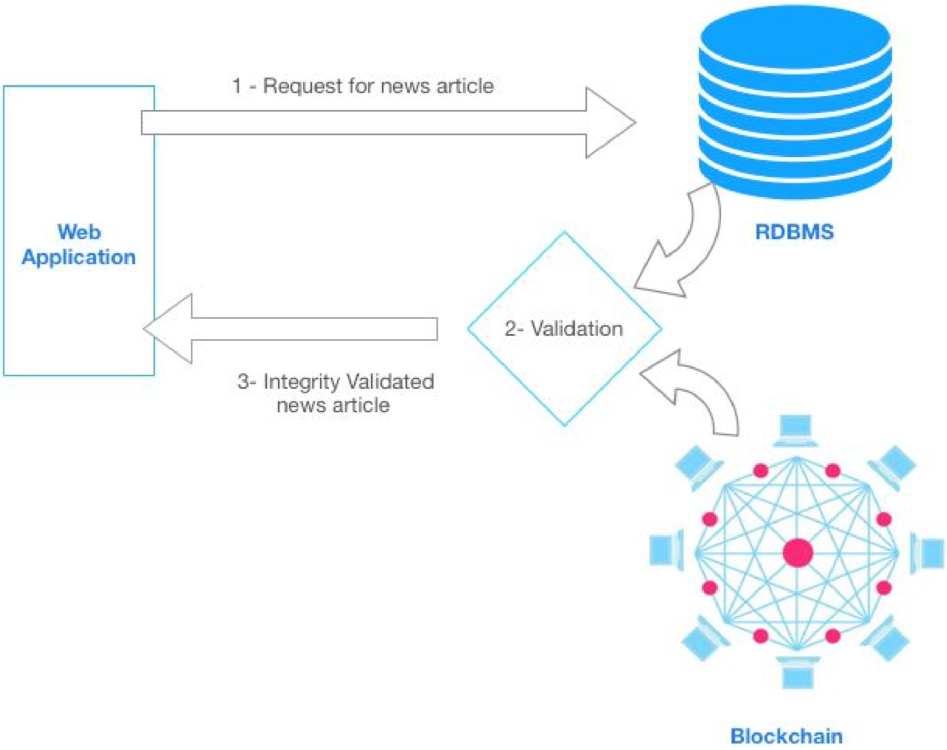

1 - Request for news article
Web Application
RDBMS
2- Validation
3- Integrity Validated news article
Blockchain

Figure 5.3: Interconnection between RDBMS and Blockchain

5.4 Authentication Model

User authentication model followed by this system is based on the challenge-response protocol (also known as Zero-Knowledge Proof) with a blockchain to store user information securely. Abstract design of the system authentication model is depicted in figure 5.4.

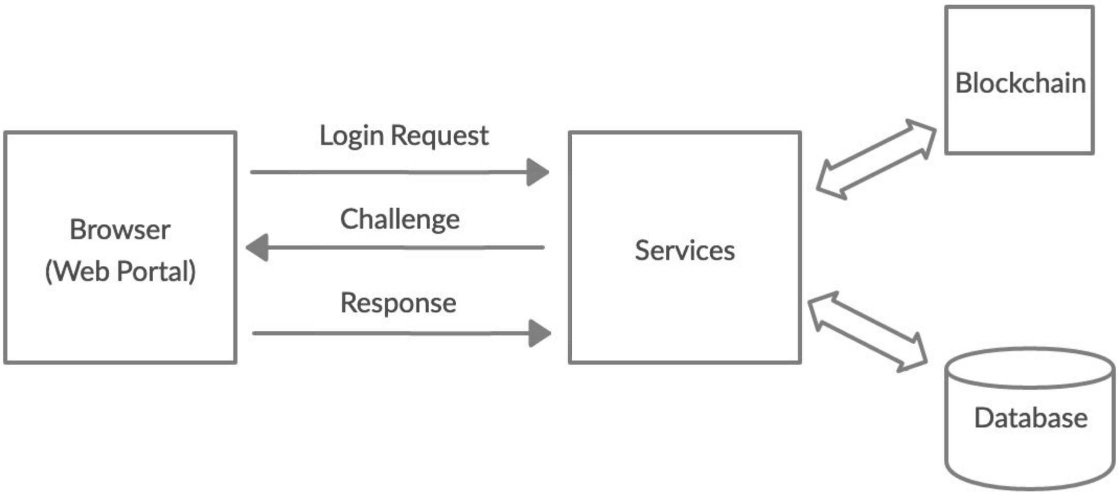


Figure 5.4: Abstract design of the system authentication model

- Blockchain keeps records of the hashes of public keys of users along with all the other user information against the user's ID (email address).
- Ballerina service uses a separate RDBMS to keep records of actual public keys and other user information.
- At the login process, the user submits the email address and the private key to the browser interface and challenge-response protocol takes place in between the service and the browser (web portal) to authenticate the user.
- User information is fetched and validated using the information stored in the RDBMS database and an SSL key pair (private-public key pair) is required to the aforementioned challenge-response protocol.

5.3 Authorization Model

Authorization to the system is designed and implemented following a simplified implementation of the W3C recommended Decentralized Identifier (DID) [25] and Verifiable Credentials (Verifiable Claims) [26].

In this case, the news reporter (i.e. Alice) can prove her identity and the authorization level (whether she has news writing permission) in the blockchain using verifiable claims with Zero-Knowledge Proof when she is entering hashed keys of her news reports into the blockchain. The verifiable claims can be issued by both governmental and non-governmental entities in the real world scenarios. But here we will assume, some higher-level 'moderator' user of the system provides these, for the convenience of demonstration of this solution. (the objective is to evaluate the possibility to use blockchains with claim based authorization, but now to exactly implement a solution with claim based authorization)

Therefore this claim based authorization acts similar to a permission model. A user profile will be created in a preferred user store repository (RDBMS) when the user initiates user registration (sign-up). In a standard implementation, in the user registration process, another service acts to retrieve verifiable claims (name, NIC, degree, date of birth, etc.) of a user (e.g. Alice) from a related authority (i.e. governmental agency). But in this implementation, a moderator-level user will directly assign specific roles (reader, writer, moderator, admin) to the user trying to register. Therefore those user roles (is one of verifiable claims) will be stored in the user profile along with other user data and also the user's public key.

These user role claim value will be considered when the user is granted access for certain operations in the system, hence the authorization is implemented in this approach. This is a simple introduction on how the user authorization is designed using blockchains with a simplified implementation of verifiable claims/claim based authentication.

Chapter 6

6. Implementation

6.1. Introduction

This research is implemented based on the hypothesis that the utilized technologies, protocols will address the problem and succeed in the particular application's authorization, authentication, non-repudiation, information integrity and credibility related use cases. The proto-type online news portal is designed and implemented with the proposing solution protocols, technologies and finally evaluation has been done on that.

6.2 System Architecture and Implementation

The prototype online news portal implemented contains 4 basic structural modules as described in chapter 5.3. (Web portal, Business Logic Layer, RDBMS database, Blockchain). This section describes how these 4 modules are implemented to achieve the expected objectives of this research.

In section 6.2, the basic structural modules of the system are described with respect to their functionality. Sections 6.3, 6.4 and 6.5, describe the functional models of the system. One or more structural modules participate collaboratively for each functional models to function to achieve the respective functionality.

6.2.1 Web Portal

6.2.1.1 Introduction

Web portal is implemented using React JS library [27] and hosted on a server started by a program written with Ballerina programming language (Ballerina v1.2.0). React JS is a popular web application framework from which it is possible to develop single-page web applications conveniently.

Some reasons behind choosing React JS are that, [28]

1. It boosts productivity and facilitates further maintenance
2. It guarantees faster rendering
3. It guarantees stable code
4. It is SEO friendly
5. It is focused and easy-to-learn
6. Has a handy developer toolset
7. Used and backed by a strong community

The web application portal is developed in a way so that the above features can be achieved. In the next sections, the most important features and concerns in developing this web application are described. They are,

1. Single page application
2. Page/component routing mechanism
3. How the web app has been served as a service on a port
4. Material-UI as UI library

6.2.1.2 Single page application

Single Page Application is a type of web application that requires only a single page to be loaded into the browser and does not require page reloading during use [29]. This is a latest trend in Web application development basically due to high performance it brings. The below code segment (Figure 6.1) illustrates how the Web application is served as a single DOM element '<App />'.

index.html:

```
…
<html>
 <body>
   ...
   <div id="root"></div>
   ...
 </body>
</html>
```

index.js:

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';

ReactDOM.render(
 <React.StrictMode>
   <App />
 </React.StrictMode>,
 document.getElementById('root')
);
```

Figure 6.1: index.js and index.html code implementation - How the web app has been served as a single page

Hence the whole web application is being served as a single web page residing under the <div> with id="root".

6.2.1.3 Component routing mechanism

Various UI views/components are rendered appropriately into the DOM via a routing mechanism. Thus, within the ‘Single-page Application’ multiples components (pages) are rendered using a React specific routing mechanism. React applications are built with React components. Therefore the routing of the view to the preferred URL location/component is achieved by using specific ‘<BrowserRouter>’,‘< Switch>’ components and ‘<Route>’ components in the App.js file. (Figure 6.2).

```
class App extends Component {
   render() {
       return (
           <BrowserRouter>
               <div>
                   ...
                   </MuiThemeProvider>
                   <Switch>
                       <Route path="/" component={Login} exact />
                       <Route path="/about" component={About} />
                       <Route path="/contact" component={Contact} />
                       <Route path="/login" component={Login} />
                       <Route path="/signup" component={Signup} />
                       <Route path="/home" component={Home} />
                       <Route component={Error} />
                   </Switch>
                   <Navigation />
               </div>
           </BrowserRouter>
       );
   }
}
```

Figure 6.2: Web application component/view routing mechanism

6.2.1.4 How the web app has been served as a service

The React web application is hosted using a service written in Ballerina programming language. How the services are deployed using the Ballerina program will be described in the next section. It is quite easy in Ballerina to write a service and expose it over a network protocol. Therefore this same mechanism is used to expose the built React web application resources on the port 8080.

Thus the news portal will be exposed on the port 8080 and URLs some pages (components) are mentioned below as an example.

● Login page	: http://localhost:8080/login
● News-view/Home page	: http://localhost:8080/home
● Use registration page	: http://localhost:8080/signup
● Contact-us page	: http://localhost:8080/contact

6.2.1.4 Material-UI as UI library

React-JS library is supported with multiple UI frameworks to easily develop stylish web pages. Blueprint, Belle, MaterialUI, React MD, React Material Bootstrap, Shards React, etc. are some of the most prominent ones. From these, MaterialUI [30] is selected as the UI framework for this web application.

6.2.2 The business logic layer

This structural module is introduced as ‘Business logic layer’ since it is the place where all the business logic resides prominently in the system. This can be introduced as the module which interconnects the UI(web interface) and the storage components(RDBMS and Blockchain) of the system. This is completely written in Ballerina programming language. There are four basic components here.

1. REST API
2. Ballerina Services implementation
3. DAO layer objects
4. Models

The business logic layer of this system has been written with Ballerina programming language. Ballerina has inherently concurrent first-class language constructs for providing and consuming services, hence it was used to implement these components.

6.2.2.1 REST API exposing services

The REST API is implemented exposing multiple services as resources to be used by the web application. Therefore, each request from the web application is sent through this REST API to access the services which facilitate doing each functionality. The REST API is exposed on the port 9090. Below is a list of prominent REST API resources implemented for various functionality.

Table 6.1: REST API Resources

Http Method	Path	Functionality
POST	/newsArticle	addNewsArticle
GET	/news-articles	newsArticlesGet
POST	/news-articles/validate	validateArticles
POST	/user	registerUser
POST	/user/approve-writer	approveWriter

POST	/login/challenge	getLoginChallenge
POST	/login/challenge-respond	loginChallengeRespond

6.2.2.2 Implementation of Ballerina Services

The operations required to do within the run-time execution of the system are exposed as REST API resources, with a Ballerina Service. Therefore, the implementation of the services has been written with the Ballerina itself.

- Ballerina Services implementation
  - A Ballerina service was developed to cater required functionalities for the previously mentioned REST API resources/operations.

- DAO layer objects
  - Layer to access the MySql database
    - AuthDAO - for all the authentication/authorization related functionalities
    - ArticleDAO - for all other article related functionalities
- Models
  - Model classes used for the convenience and efficiency of the program execution : User, Role, NewsArticle, UserSession, ArticleValidity, NewsServiceErrorData

6.2.3 Database implementation

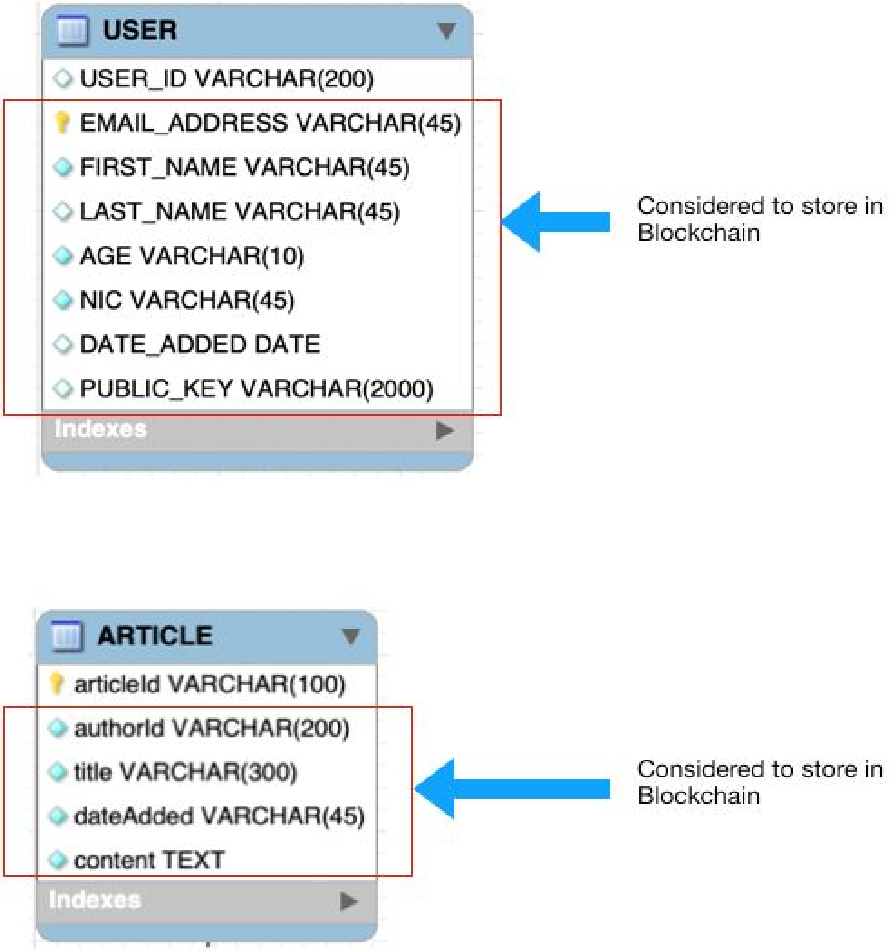
The RDBMS database is implemented with MySql 5.7. The tables in the RDBMS are,

1. USER
2. USER_ROLE
3. USER_ROLE_MAPPING
4. USER_PERMISSIONS

5. ROLE_PERMISSION_MAPPING
6. ARTICLE
7. IMAGE

Two tables among these are related to the data stored in the blockchain. They are ‘USER’ and ‘ARTICLE’ tables.

Hash values of the entries (rows) in USER table and ARTICLE table are stored in the blockchain separately as individual transactions for each entry. Figure 6.3 illustrates which columns(cells in a row) are considered to be stored in the blockchain and which are not.



USER
USER_ID VARCHAR(200)
EMAIL_ADDRESS VARCHAR(45)
FIRST_NAME VARCHAR(45)
LAST_NAME VARCHAR(45)
AGE VARCHAR(10)
NIC VARCHAR(45)
DATE_ADDED DATE
PUBLIC_KEY VARCHAR(2000)

ARTICLE
articleId VARCHAR(100)
authorId VARCHAR(200)
title VARCHAR(300)
dateAdded VARCHAR(45)
content TEXT

Indexes
Considered to store in Blockchain
Indexes
Considered to store in Blockchain

Figure 6.3: Data considered to be stored in Blockchain

According to Figure 6.3, the ID values of both tables are not considered to hash and store in the blockchain. It is because both articleId and userId contain the blockchain transaction ID of each entry, so they cannot be stored in the blockchain. These columns act as the mapping entry in-between the blockchain entries and the MySql rows.

When generating a hash value, following steps are followed. (this is mentioned taking the Article table as an example)

1. Data in the columns except articleId is taken in JSON format.
2. Sha256 hash of the JSON string is generated.
3. Base16 encode the above-generated hash

The Base16 encoded value is sent into the blockchain within the parameters, (in "data" param) while making the transaction. This is done by invoking the Ethereum JSON RPC method "eth_sendTransaction". This method call returns the blockchain transaction ID and it will be stored as the articleId in the database in the Article table.

Similarly, the userId will be stored in the User table after storing user info hash in the blockchain via a transaction.

6.2.4 Blockchain Setup

The blockchain setup is deployed using Go Ethereum (version 1.9.11-stable), usually abbreviated to Geth. Geth is a mature implementation of the Ethereum node software.

A number of nodes (miners) in the blockchain will be determined by the number of news writers available in the system. The proposal is that each news writer will be a miner, hence will maintain a blockchain node. This is assuming that all the writers are employees in the news portal organization.

The organization can decide the number of miners as prefered based on the level of security needs to achieve, infrastructure availability, cost affordable, etc. Higher the number of nodes, higher the level of security the blockchain achieves.

Each miner (news writer) will have their own ethereum account. Accounts are essential for users to interact with the Ethereum blockchain via transactions. It is also possible to maintain multiple accounts and miners representing news readers so that read operations only will be done via those accounts.

6.3 Implementation of News article writing-reading model

News article writing-reading model is designed in a way that tampering the news article content is highly impossible, so that the integrity of the content is preserved. The basic flow of this process is described in Figure 5.1. How news article data is stored in the MySql database and the blockchain is described in section 6.2.3. Hence this is a description on how each of these sub-processes takes place in the processes of news article writing and reading in the prototype application system. Figure 6.4 is an illustration of the architecture of news article writing and reading model, including the prior mentioned processes.

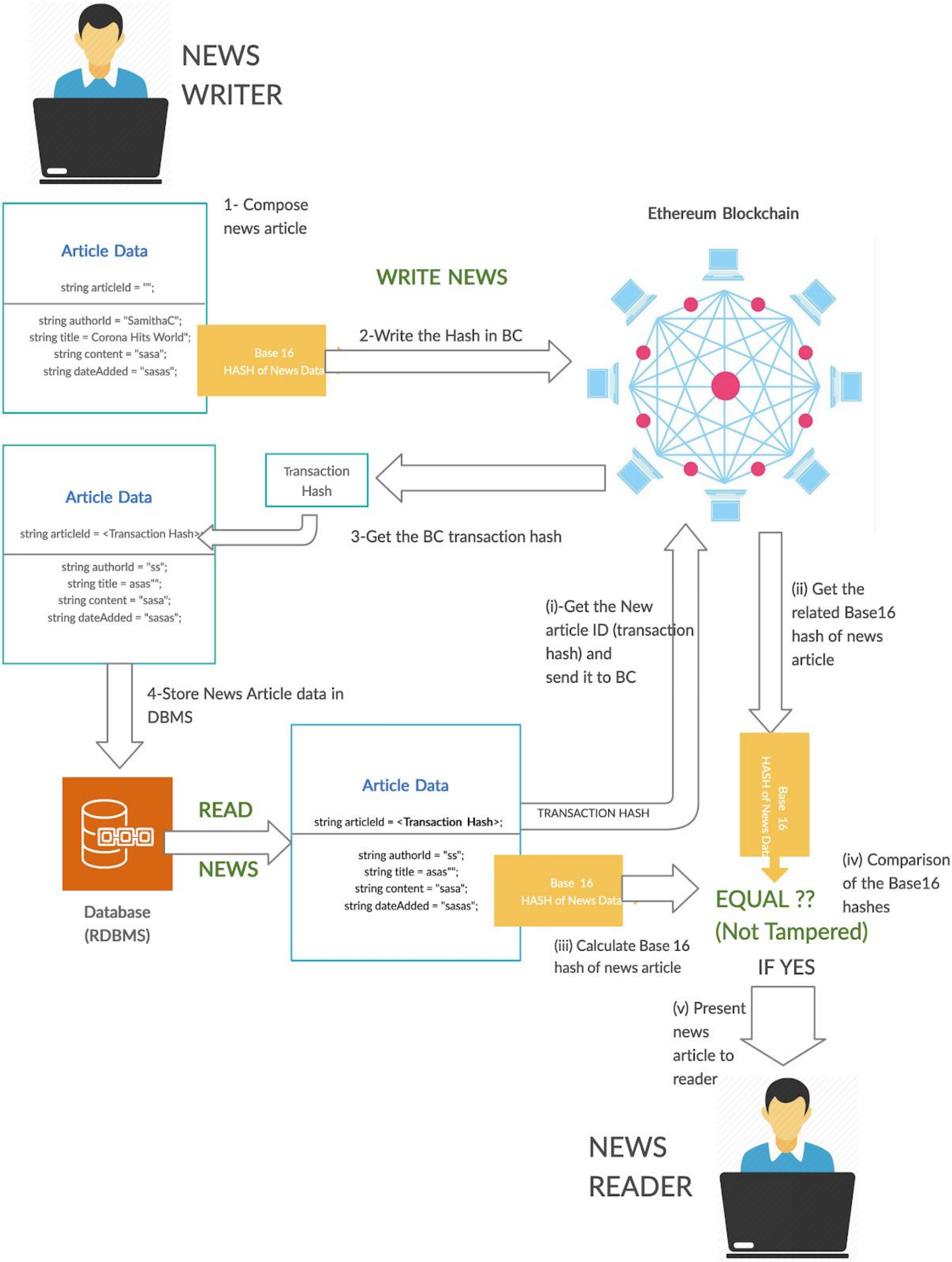

NEWS WRITER
1- Compose news article
Article Data
string articleId = "";
string authorId = "SamithaC";
string title = Corona Hits World";
string content = "sasa";
string dateAdded = "sasas";
Base 16 HASH of News Data
WRITE NEWS
2-Write the Hash in BC
Ethereum Blockchain
Transaction Hash
Article Data
string articleId = <Transaction Hash>;
string authorId = "ss";
string title = asas"";
string content = "sasa";
string dateAdded = "sasas";
3-Get the BC transaction hash
4-Store News Article data in DBMS
(i)-Get the New article ID (transaction hash) and send it to BC
(ii) Get the related Base16 hash of news article
Database (RDBMS)
READ NEWS
Article Data
string articleId = <Transaction Hash>;
string authorId = "ss";
string title = asas"";
string content = "sasa";
string dateAdded = "sasas";
TRANSACTION HASH
Base 16 HASH of News Data
Base 16 HASH of News Data
(iii) Calculate Base 16 hash of news article
(iv) Comparison of the Base16 hashes
EQUAL ?? (Not Tampered)
IF YES
(v) Present news article to reader
NEWS READER

Figure 6.4: Architecture of News Article writing and reading model

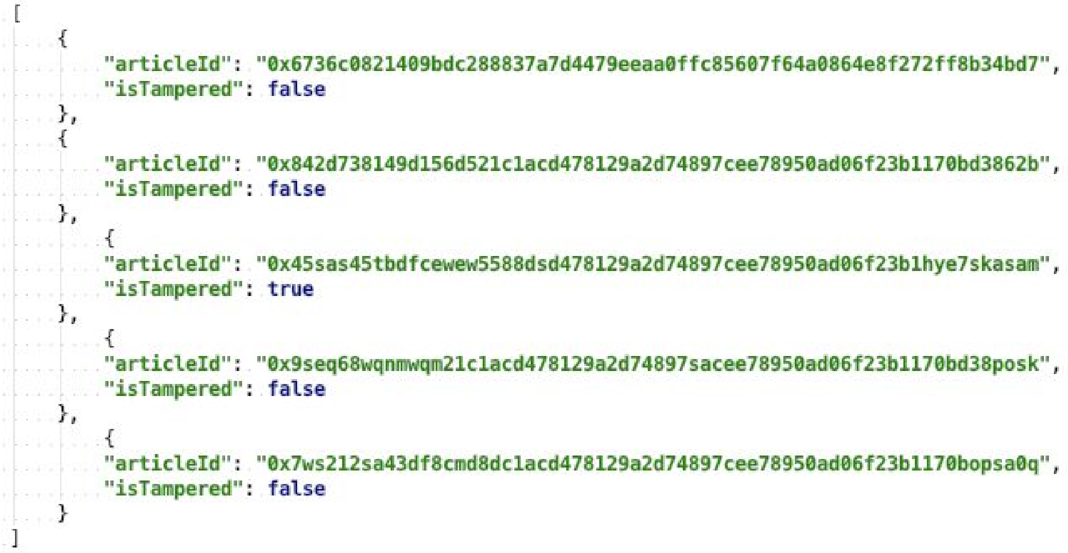
Writing an article

1. When a user who has “Writer” role login into the news portal he can fill in the news article content (title, content, etc) on the dedicated news article submission page.
2. Once he submits the article the web app will invoke the related REST API resource on the ballerina service (POST /addNewsArticle)
3. The request will be brought forward to the implementation layer from where the required operations are implemented.
4. Submitted news article information will be converted into the Ballerina record NewsArticle (model).
5. JSON string representation will be generated by the news article record and it will be encoded with the Sha256 hash function.
6. The encoded value will be encoded back to a Base 16 string.
7. The encoded value will be sent over to the blockchain in transaction payload.
8. Transaction will return a hash value which is identified as the transaction hash.
9. Transaction hash is stored as the “articleId” in the NewsArticle record. (articleId was empty until now)
10. The news article information (with the articleId) is inserted into the ‘Article’ MySql table

Reading an Article

1. When a user who has “reader” role, login to the news portal, a request is sent to the MySql database to select all the articles.
2. Article information is sent to the web portal and they are rendered on the web page.
3. Meanwhile, the same set of article information is sent to the ballerina service for validation (integrity validation). (The below steps 4 to 10 are executed to all the articles selected).

4. For each article, the article information will be converted into the Ballerina record NewsArticle (model).
5. ‘articleId’ attribute is removed from the news article record.
6. JSON string representation is generated by the news article record and it is then encoded with Sha256 hash function.
7. The encoded value is encoded back to a Base 16 string. Let’s call this encoded value as “New Hash Value”.
8. Now a JSON RPC request (eth_getTransactionByHash) is sent to the Ethereum blockchain to get transaction details by the article hash (articleId) which was fetched from the database in step 1.
9. It returns the blockchain transaction hash of the related news article. Let’s call this hash as “Old Hash Value”
10. Now, these two hashes are compared. If the hash values are different it can be decided that the article information is tampered/changed in the MySql database (or in the Blockchain which is highly impossible and rare). If the hashes are identical, no tampering has taken place.
11. This is done for each article and the result is sent to the Web portal in JSON format similar to Figure 6.5.
12. Web portal marks the articles which are tampered in the news articles - view page. Articles which are not tampered (valid articles) will be displayed normally.



```
[
    {
        "articleId": "0x6736c0821409bdc288837a7d4479eeaa0ffc85607f64a0864e8f272ff8b34bd7",
        "isTampered": false
    },
    {
        "articleId": "0x842d738149d156d521c1acd478129a2d74897cee78950ad06f23b1170bd3862b",
        "isTampered": false
    },
        {
        "articleId": "0x45sas45tbdfcewew5588dsd478129a2d74897cee78950ad06f23b1hye7skasam",
        "isTampered": true
    },
        {
        "articleId": "0x9seq68wqnmwqm21c1acd478129a2d74897sacee78950ad06f23b1170bd38posk",
        "isTampered": false
    },
        {
        "articleId": "0x7ws212sa43df8cmd8dc1acd478129a2d74897cee78950ad06f23b1170bopsa0q",
        "isTampered": false
    }
]
```

6.5: Article validity response

6.4 Implementation of User information Storage Model

The storage model of the user information is similar to the news article storage model. User information is stored in the MySql database and a hash value of user information is stored in the ethereum blockchain. When fetching the user information in the process of user authentication/authorization, the hash value of the user information is generated again and compared with the related hash value stored in the blockchain. Figure 6.6 illustrates the user information storage model with each basic step.

The basic steps carried out in new user registration (sign-up) are described below.

1. User submits the user information to self-register into the system. As depicted in Figure 6.6 ‘userId’ and the ‘roles’ information are still not available. Other information is used to map to a Ballerina User record entry and the information is inserted into the database, once the user submits the form.

2. Then another user who has ‘admin’ or ‘moderator’ roles will assign roles to the user via a separate portal. This information will be directly inserted into the database.
3. Now the Ballerina User record entry has values for all the attributes, except for “userId”. This record entry is converted into a JSON formatted string, and it will be encoded with Sha256 algorithm.
4. The Sha256 encoded value will be converted into base16 string representation.
5. The encoded value will be sent over to the blockchain in transaction payload.
6. Transaction will return a hash value which is identified as the transaction hash.
7. Transaction hash is inserted into the ‘User’ table in the MySql database as the “userId” in the related row. (userId was empty until now)

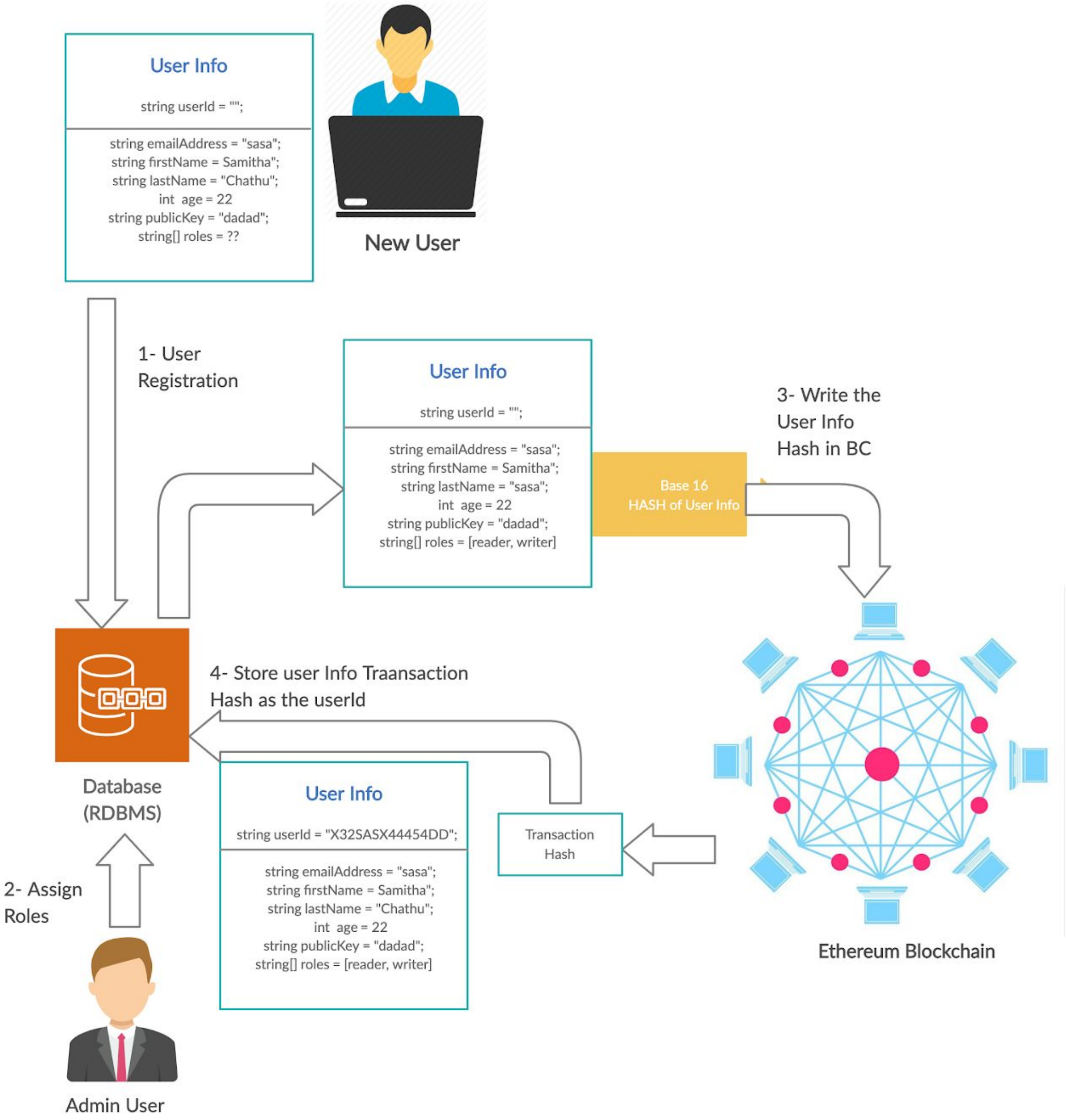

User Info
string userId = "";
string emailAddress = "sasa";
string firstName = Samitha";
string lastName = "Chathu";
int age = 22
string publicKey = "dadad";
string[] roles = ??
New User
1- User Registration
User Info
string userId = "";
string emailAddress = "sasa";
string firstName = Samitha";
string lastName = "sasa";
int age = 22
string publicKey = "dadad";
string[] roles = [reader, writer]
Base 16
HASH of User Info
3- Write the User Info Hash in BC
Database (RDBMS)
4- Store user Info Traansaction Hash as the userId
User Info
string userId = "X32SASX44454DD";
string emailAddress = "sasa";
string firstName = Samitha";
string lastName = "Chathu";
int age = 22
string publicKey = "dadad";
string[] roles = [reader, writer]
Transaction Hash
Ethereum Blockchain
2- Assign Roles
Admin User

Figure 6.6: Implementation of User info Storage Model

6.5 Implementation of User Authentication Model

Figure 6.7 illustrates the sequence diagram of this process.

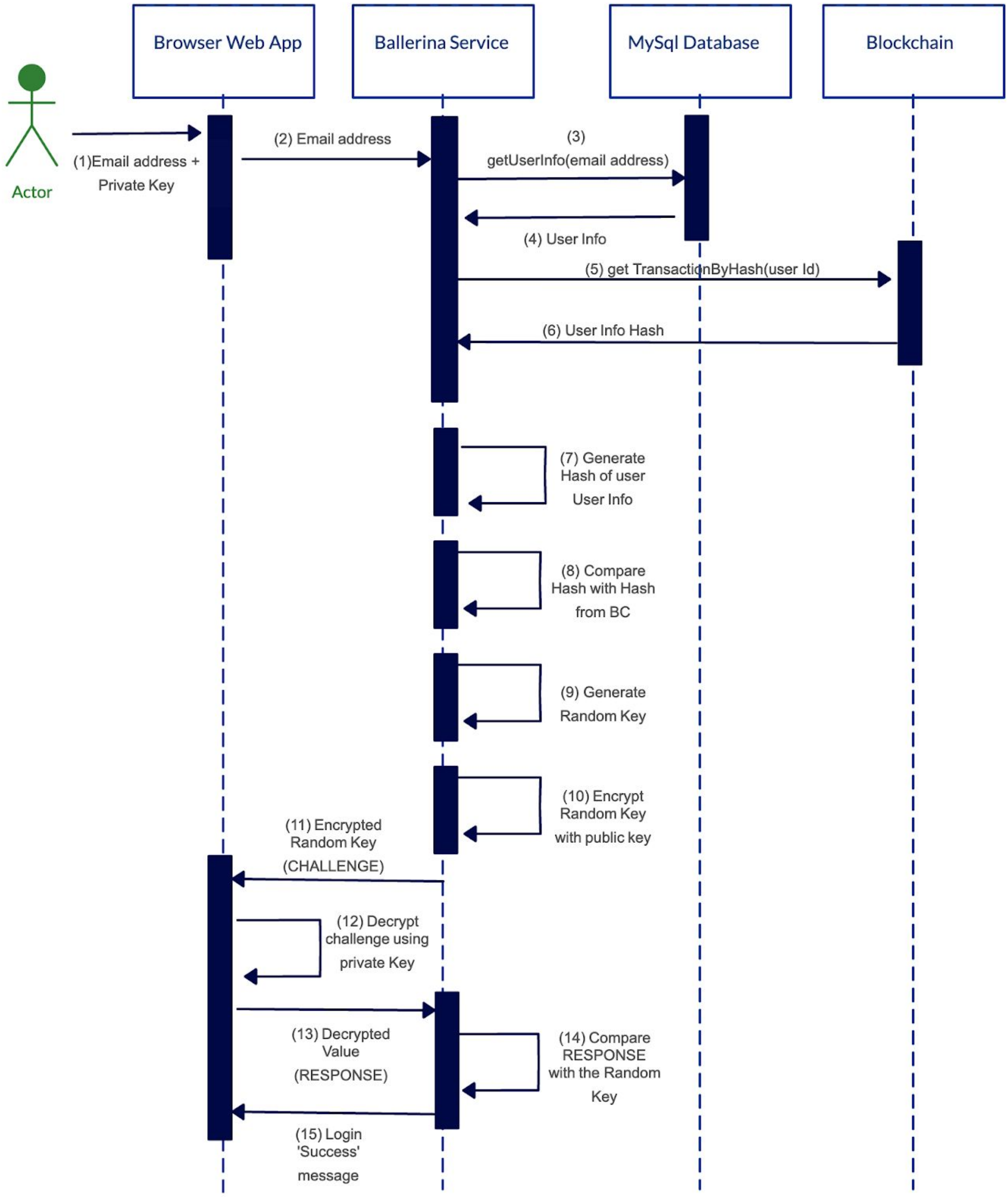

Browser Web App
Ballerina Service
MySql Database
Blockchain
Actor
(1)Email address + Private Key
(2) Email address
(3) getUserInfo(email address)
(4) User Info
(5) get TransactionByHash(user Id)
(6) User Info Hash
(7) Generate Hash of user User Info
(8) Compare Hash with Hash from BC
(9) Generate Random Key
(10) Encrypt Random Key with public key
(11) Encrypted Random Key (CHALLENGE)
(12) Decrypt challenge using private Key
(13) Decrypted Value (RESPONSE)
(14) Compare RESPONSE with the Random Key
(15) Login 'Success' message

Figure 6.7: Sequence Diagram: Implementation of User Authentication Model

Challenge-response protocol has been implemented for the user authentication process for which the React Web application, Ballerina service, MySql database and the Ethereum blockchain participate actively.

Figure 6.7 illustrates the sequence diagram of this process. Each step is described in detail below. All these steps are executed in a synchronous flow.

1. User submits the user's email address and the private key in the login page. A screenshot of the login page can be found in Appendix A- Figure 1.
2. Once the user submits the credentials, the browser sends the email address to the ballerina service layer. The REST API resource invoked for this is "POST /login/challenge"
3. Ballerina service query the MySql database to get the user information for the given email address.
4. Database returns the user information.
5. Ballerina service then extracts the userId (user hash) from the returned user information and sends JSON RPC call "eth_getTransactionByHash" to the ethereum blockchain to get the blockchain transaction bound to the userId.
6. Blockchain returns the user information hash value.
7. Ballerina service maps the user information into Ballerina User record entry. It removes the userId value from the record and converts the information into a JSON string representation. This JSON string is then encoded in SHA256 function and that value will be encoded into a Base16 string value. This value is the "New hash value" of the user information.
8. The "New hash value" generated in step 7 is compared with the userId (user hash) which was extracted from the database result in step 5. If the two are similar it determines that the user information has not been tampered. But if they are not similar, the decision is taken that the user information is tampered either in the database or the blockchain. (tampering in the blockchain is highly impossible). If the decision is that the user information is tampered, the authentication process

fails and halts there.

9. If no tampering has taken place, the ballerina service generates a Random key of 12 characters.
10. Then that random key is encrypted with the user's public key. This encrypted value is called the "Challenge".
11. The challenge is returned to the browser (web portal).
12. Web application decrypts the challenge using the private key submitted by the user.
13. Web application makes another REST API call to the ballerina service resource "POST /login/challenge-respond", with the decrypted challenge (RESPONSE) as the payload.
14. Ballerina service now compares this sent in the payload with the "Random key" generated in step 9. If they are similar, user authentication (login) is a success. Else it is failed.
15. Ballerina service returns the response containing the information whether the user authentication is 'success' or 'failed'.

According to the above process user information stored in the database is validated for any tampering, using the blockchain involvement. (If they are tampered, user login fails.) Thus the integrity of user information including the user roles and public key are preserved by this approach. Therefore, the challenge-response protocol implemented in this system is strengthened with another layer of security; "integrity check". If the user information integrity is violated, the user is unable to get authenticated into the system.

6.6 Implementation of User Authorization Model

When a user registers into the system his information is stored in the MySql database. After that, a moderator/admin user will assign required roles to the user. For example, if the user is eligible to get the 'writer' role, the moderator user grants him that role by inserting the relevant role mapping into the database. (this process is introduced as ‘approving’ the user). Once the user is approved, all the user information, i.e. roles, public key, email address, etc. will be taken and will go through the below steps.

1. Data in the columns except userId is taken in JSON format.
2. Sha256 hash of the JSON string is generated.
3. Above generated hash is Base16 encoded.

This base16 encoded hash of all the user information is stored in the blockchain as a transaction via the transaction payload.

Therefore assume, when Alice logins into the system, the backend service will retrieve her user profile hash from the blockchain (recorded against email), then retrieve the user profile from the user store in RDBMS. It extracts the user roles from the profile data. This user “roles” is identified as a user claim. Then it will check whether her roles include particular entries (e.g. Writer role) to determine whether she has write-access to the news system. The backend service also validates the integrity of the user information by regenerating the user hash and comparing it with the one already stored in the RDBMS. The authorization process takes place just after the authentication process, hence the user information validation step is common to both the authorization and authentication.

Therefore based on the verified role claim of the user, the ballerina service allows/disallows access to certain resources in the Ballerina service. If the user “roles” are tampered in the database, it is detected by the hash-comparing process. This is a simplified claim-based authorization process, because it uses user claim values to authorize the user. Since these user claims are protected and made immutable by the participation of blockchain, we can identify this simplified claim-based authorization process as strengthened rather than the standard claim-based authorization process.

Chapter 7

7. Evaluation

7.1 Introduction

This research is conducted with the expectation of fulfilling the following objectives.

- Implementing a solution to overcome the problem of assuring immutability of audit trail critical systems using blockchain and related technologies after a critical study of the problem and literature.
- The research is done based on the hypothesis that the to be utilized technologies, protocols will address the problem and succeed in assuring the immutability of data/records meanwhile establishing secured authentication and authorization into the system.
- A proto-type application is designed and implemented with proposing solution Protocols and technologies.

Therefore the evaluation of this study is conducted to evaluate whether the above-mentioned objectives are achieved within the specified scope of ensuring immutability of data/records, of the online news portal; using blockchain technology.

7.1 Evaluation Approach

Approaches followed to evaluate this research are listed below.

1. Evaluation of the functionality
2. Demonstration of security breach attempts
3. State transition diagrams
4. Comparison with existing solutions

The evaluation was done in an environment as follows.

- Browser used to access the web portal: Chrome Version 83.0
- Operating System : MacOs v10.14.6
- Blockchain nodes are running in the same OS
- Used DB client: MySql Workbench 8.0
- Used REST API Client : Postman 7.26.0

7.2.1. Evaluation of the functionality

When evaluating the functionality, the following operations are tested to verify whether they are successfully implemented.

- Login to the news portal
- Logout from the news portal
- Assign role claims to the users by the admin/moderator user
- Registration of a new account(user) in the system.
- Authorization level (permission level) for news reader/ writer
- Adding news articles and publishing them
- Viewing news articles
- Information backup/recovery (when one or more blockchain miners down)

It was possible to do the above operations without any issue as expected.

7.2.2. Demonstration of security breach attempts

Following tests were done to demonstrate particular security breach attempts. Basically login (user authentication), user authorization and information immutability related tests were done to demonstrate the achievements of each aspect, as expected.

- Login without having the correct private key
    - Test : Submit an incorrect/unrelated private key to log in.

- Result: Passed; Login failed as expected

- Login into the system by tampering SSL key pairs in DBMS
  - Test: Replace the public key of a user in the DBMS with another public key (this is the hacker's public key) and try to login into the system with the hacker's private key.
  - Result : Passed; Login failed as expected. This fails when the hash of the user information (including public key) stored in the DBMS is compared with the hash stored in the blockchain..

- Add news articles by an unauthenticated user
  - Test: Tried to add a news article by a user with an incorrect private key.
  - Result: Passed ; user login phase itself failed and so cannot at least access the news article adding page

- Add news article by an unauthorized user
  - Test: Tried to add an article by login via "reader" user (user with only "reader" role)
  - Result: Passed ; Cannot submit the news article
- Edit/change the content of new articles through different layers.
  - Through application itself (web portal)
    - Editing the news articles is not supported through the web application as of now. Thus there is nothing to test here
  - Through the exposed REST API endpoints
    - Editing the news articles is not supported through the REST API as of now.
    - Result: Passed; Cannot update since there is no REST API provided for this
  - By INSERT/UPDATE queries made on the database

    - Test: Authorized user (i.e. admin) changes the news article content by a simple update query executed on the "ARTICLE" table.
    - Result: Passed ; Changed article was displayed as "Tampered" in the news article viewing page. Therefore if there is an integrity violation in the system data, it is detected and displayed to the users just after the violation takes place.
- ○ Attempts to dispute data in the blockchain
    - Mutating data in a blockchain is a whole big effort and technically identified as highly difficult. Even one needs to have the related technological knowledge and expertise and processing power to tamper blockchain data. Therefore no effort was made to do this. Immutability/tamper-resistance is the core attribute of a blockchain and it is designed to be not possible. Hence not required to evaluate this in this research.

7.2.3. Logical Analysis with State transition/ flow diagrams

This approach of evaluation is a type of logical evaluation method. We consider the flow, state-transitions of each process to determine that the expected flow of events/actions will take place without any issue and will not break the flow.

The flow of each of the basic processes of the system; user authentication, news article reading-writing model, user information storage model are illustrated by the diagrams Figure 6.7, Figure 6.4 and Figure 6.6 respectively. Hence when analyzed logically, there is no possibility of breaking the expected flow of each process.

User authentication will proceed and succeed if the user exists and the submitted private key is valid and no information tampering has taken place both in the database and the blockchain. If any of the above conditions are false the user authentication fails and user is requested to submit correct credentials. An information tampering has been taken place, the user will be informed on that and hence the system admin will need to take necessary actions to remediate the tampered credentials of the user.

In the same way, user authorization process will proceed and succeed (after user authentication is succeeded) if the user's role claims are not tampered in the database and if the user has correct roles expected. If any of the above conditions are false the user authorization fails and user will not be granted the access as expected.

News information writing and reading model will work as expected as in Figure 6.4. If the user has successfully authenticated and authorized, he will be able to publish a news article without an issue. When reading the article content (loading articles in the web portal), they will be displayed correctly, but if the article content is tampered, they will be displayed with a clear indication that the article is tampered(integrity validation failed).

Therefore according to this logical analysis of the activity flow/state transitions it can be stated that the proposed solution system will work smoothly as expected without getting blocked at any state.

7.2.4. Comparison with existing solutions

The proposed solution can be compared with the already existing solutions which were discussed in chapter 2. The positives of the solution are described initially below.

- The solution approach is not highly complex.

The followed design and the protocols are conveniently applicable for any other audit history critical system. The approaches suggested in other literature are mostly highly complex.

“The News Provenance Project” [6] is suggesting to establish a whole new protocol to carry a set of specific signals/metadata that can travel with published media anywhere that material is displayed. Some other studies suggest converting traditional RDBMS into an RDBMS in which the data delete/modify operations are removed [9]. Even there are research done to invent immutable databases from scratch. i.e. Datomic, Mentat [10] and storing all the historical data instead of updating database cells. Even, some researchers have come up with solutions with multiple blockchain layers [13].

- Relatively better additional overhead added by the blockchain.

Accessing data from a blockchain is slower than accessing from a database. But the system proposed in this solution fetches news article data primarily from the database and the validation happens only in the background. Therefore, the participation of blockchain doesn’t affect the news article information loading time. (but there is an effect on user authentication and authorization processes suggested since they cannot be done asynchronously). In another study, two layers of blockchains have been utilized. Data is accessed from PoW-less blockchain, hence latency is not increased much.

In another study, a second layer of blockchain has been exploited to act as an access control layer [14], but the data is fetched from the RDBMS upon successful authorization from the blockchain. This approach is anyway vulnerable to data integrity violations in the RDBMS itself.

- Relatively less overall cost.

Because the suggested system is less complex, the infrastructure to be utilized costs, less than other complex systems. Maintenance cost also will be less.

- Does not deviate from standard technologies and software

The suggested system doesn’t use modified RDBMS or specifically developed RDBMS. It can work with any kind of DBMS. Hence this solution can be applied with other enterprise applications without making the system incompatible with other standard applications.

- The blockchain has been utilized for the security of authentication/authorization layers too.

Because of this, the guarantee on the integrity of information is improved also via these layers of the application. A similar approach is used in [12] with challenge-response protocol based authentication, but it has used smart contracts instead of transactions to store information.

Chapter 8

8. Conclusion & Further work

8.1 Overall Achievement

According to the result of the evaluation approaches followed as in the previous chapter, it is possible to state that the prime aim of the study is achieved. It was possible to come up with a system which has improved assurance of information/data immutability and integrity in this audit history critical online news portal.

8.2 Limitations of the Solution Presented

Considering this prototype application domain, (news portal), ability to update already published articles legitimately, is a non-trivial requirement, even though this is specified as out of scope in this study. Therefore, considering the prototype application domain this can be stated as a limitation of the system. But with the provided system, implementing the ability to update content should be doable. This is further described in the 'Further Work' section.

Securing the pictures/photos/graphical-content of articles is not implemented in the system. Even though they can be stored in the RDBMS and displayed on the currently developed news portal, with respect to each article, they are not secured via blockchain. In other words, a hash value of the graphical content is not stored in the blockchain. Hence this also can be stated as a limitation of the system. But it is possible to store/secure any type of data, in blockchain as similarly done for text data.

8.3 Achievement of the objectives

The judgement of the objectives with the outcomes of the evaluation is vital for final interpretation of the research conclusion. Hence this section considers all the objectives of the research which were stated in chapter 1.4.2 to conclude whether they have been achieved.

The major objective was stated as designing a solution to overcome the problem of assuring immutability and integrity of audit trail critical systems using blockchains. This is achieved according to the outcomes of evaluation of the functionality and demonstration of security breach attempts. All the expected functionality operations could be performed and the security breach attempts related to information integrity were failed. (so integrity was not compromised).

The next objective was designing a solution to improve the security of user authentication process of a system, concerning on preserving the integrity of information used for the authentication process. This also is achieved since we could demonstrate that tampering of SSL public key stored in DBMS cannot pave the way to illegally login into the system. Therefore, the challenge-response protocol based authentication is successfully improved by the proposed approach.

It was also demonstrated that unauthorized user cannot submit news articles. Simplified implementation of the claim-based authorization was successful. It was supported by blockchains by storing hashes of user information (which includes user's role values) in the blockchain. Therefore, it is obvious that the process is improved by the methodology followed, hence the 3rd objective is achieved.

The final objective was generally to apply the above achieved objectives into a real-world audit-history critical application. According to the evaluation of the functionality and demonstration of security breach attempts, it can be stated that the underlying objective too was achieved.

8.2 Further Work

The developed system doesn’t allow to update the content of articles already added. This was not implemented since the immutability preservation was the basic requirement of this study. But there can be legitimate requirements to update a news article after publishing them. The author itself would need to update the article just after publishing, maybe to correct some typos, to add some additional information, to remove irrelevant wordings in the article, etc. Hence this feature can be added as further work. Anway it will require a separate blockchain transaction to store the updated news article content (hash) because it is not possible to update a transaction in a blockchain. Therefore, this will be similar to a version management mechanism. Older versions of the same article will not be displayed on the news portal, but only the latest one will be displayed. In this case, it will be vital to indicate that the document is updated, and information about the update (who updated, when, etc.) on the article viewing page of the web portal, so that the integrity of the article is not impaired.

References

[1] Security Intelligence. 2016. Sabotage: The Latest Threat To The Financial/Banking Industry. [online] Available at: https://securityintelligence.com/sabotage-the-latest-threat-to-the-financialbanking-industry [Accessed 16 March 2020].

[2] IT Security Guru. 2016. 2017 The Year Of The Data Integrity Breach. [online] Available at: https://www.itsecurityguru.org/2016/11/29/2017-year-data-integrity-breach [Accessed 6 April 2020].

[3] J. Uchill, "Hackers selling access to tamper with U.S. news sites," *Axios*, 15, January, 2019. [Online]. Available: https://www.axios.com/russianhackers-us-news-site-access-b93ab118-72d3-4da4-9c7f-823f92e7a7bf.html [Accessed: 24 January, 2020].

[4] Fischer, S., 2017. When Hackers Target The News. [online] Axios. Available at: https://www.axios.com/when-hackers-target-the-news-1513301494-3ae754f2-b1ea-4ffc-9b30-5a85f8e275a1.html [Accessed 23 May 2020].

[5] S. Koren, "Introducing the News Provenance Project," *NYT Open*, 23 July, 2019. [Online]. Available: https://open.nytimes.com/introducing-the-news-provenance-project-723dbaf07c44#ae53 [Accessed: 24 January, 2020].

[6] "The News Provenance Project." [Online]. Available: https://www.newsprovenanceproject.com/about [Accessed: 24 January, 2020].

[7] S. Perera, "Reality Check: Evaluating Blockchain for Enterprise Integration," *WSO2 Research Blog*, 22-May-2019. [Online]. Available: https://wso2.com/blog/research/evaluating-blockchain-for-enterprise-integration [Accessed: 24 January, 2020].

[8] S. Perera, “Four architecture pattern candidates for Blockchain-based decentralized applications,” *FreeCodeCamp*, 23 April, 2019. [Online]. Available: https://www.freecodecamp.org/news/https-medium-com-srinathperera-blockchain-patterns-6cf58fdc2d9b [Accessed: 24, January, 2020].

[9] M. W. B. Duncan, “Creating an Immutable Database for Secure Cloud Audit Trail and System Logging,” in *Special Track running alongside CLOUD COMPUTING 2017*, Athens, Greece.

[10] T. Stauber and S. Keller, "Immutable databases at the example of Datomic", University of Applied Sciences Rapperswil, 2016.

[11] Y. Challal, “Federated Claims Based Authentication and Access Control in the Vehicular Networks,” presented at the Network and Information Systems Security, La Rochelle, France.

[12] M. Dayarathna, “Challenge Response Protocol Based Authentication with Blockchains,” 17 April, 2019. [Online]. Available: https://medium.com/@miyurud/challenge-response-protocol-based-authentication-with-blockchains-2935bf6c53d4 [Accessed: 24 JAnuary, 2020].

[13] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri and V. Sassone, "Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments", in *Italian Conference on Cybersecurity*, 2017.

[14] Samantha Tharani, J., Tharmakulasingam, M. and Muthukkumarasamy, V., 2020. A blockchain-based database management system. *The Knowledge Engineering review*, [online] 35(20). Available at: https://www.cambridge.org/core/journals/knowledge-engineering-review/article/blockchainbased-database-management-system/9F946ACEB1041D6B075F593ABE024BDF/share/5a4ea1ea08d8ef2a7c9f02295b76fb45d927f5d4 [Accessed 16 June 2020].

[15] J. Zhang, S. Zhong and T. Wang, "Blockchain-based Systems and Applications: A Survey", Journal of Internet Technology, vol. 21, no. 1, 2020.

[16] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, “A Survey of Blockchain Applications in Different Domains,” Proceedings of the 2018 International Conference on Blockchain Technology and Application - ICBTA 2018, 2018.

[17] Junestrand, S., 2019. What Is Blockchain?. [online] Stefan Junestrand. Available at: https://www.stefanjunestrand.com/blog/what-is-blockchain [Accessed 20 May 2020].

[18] Rosic, A., 2016. What Is Blockchain Technology? A Step-By-Step Guide For Beginners. [online] Blockgeeks. Available at: https://blockgeeks.com/guides/what-is-blockchain-technology [Accessed 4 February 2020].

[19] "Zero-knowledge proofs explained: Part 1", *ExpressVPN*. 2020 [Online]. Available: https://www.expressvpn.com/blog/zero-knowledge-proofs-explained/. [Accessed: 29- Mar- 2020]

[20] Bhardwaj, C., 2020. *What Is Zero-Knowledge Proof & Its Role In The Blockchain World*. [online] Appinventiv. Available at: https://appinventiv.com/blog/zero-knowledge-proof-blockchain [Accessed 29 March 2020].

[21] Trivedi, J., 2020. *Claim Based And Policy-Based Authorization With ASP.NET Core 2.1*. [online] c-sharpcorner. Available at: https://www.c-sharpcorner.com/article/claim-based-and-policy-based-authorization-with-asp-net-core-2-1 [Accessed 10 March 2020].

[22] Krawczyk, T., 2015. *Introduction To Claims-Based Authentication And Authorization In .NET*. [online] Future Processing. Available at: https://kariera.future-processing.pl/blog/introduction-to-claims-based-authentication-and-authorization-in-net [Accessed 1 June 2020].

[23] L. Marx, 2018. *Storing Data On The Blockchain: The Developers Guide*. [online] Malcoded. Available at: https://malcoded.com/posts/storing-data-blockchain [Accessed 25 March 2020].

[24] W. Choy, "Blockchain 51% Attacks – Lessons Learned for Developers and Trading Platform Operators", *National Law Review*, 2020 [Online]. Available: https://www.natlawreview.com/article/blockchain-51-attacks-lessons-learned-developers-and-trading-platform-operators. [Accessed: 02- Jun- 2020]

[25] "Decentralized Identifiers (DIDs) v1.0." [Online]. Available: https://w3c.github.io/did-core [Accessed: 24 January, 2020].

[26] "Verifiable Credentials Use Cases." [Online]. Available: https://www.w3.org/TR/vc-use-cases [Accessed: 24 January, 2020].

[27] J. Walke, *React JS*. Facebook, 2013.

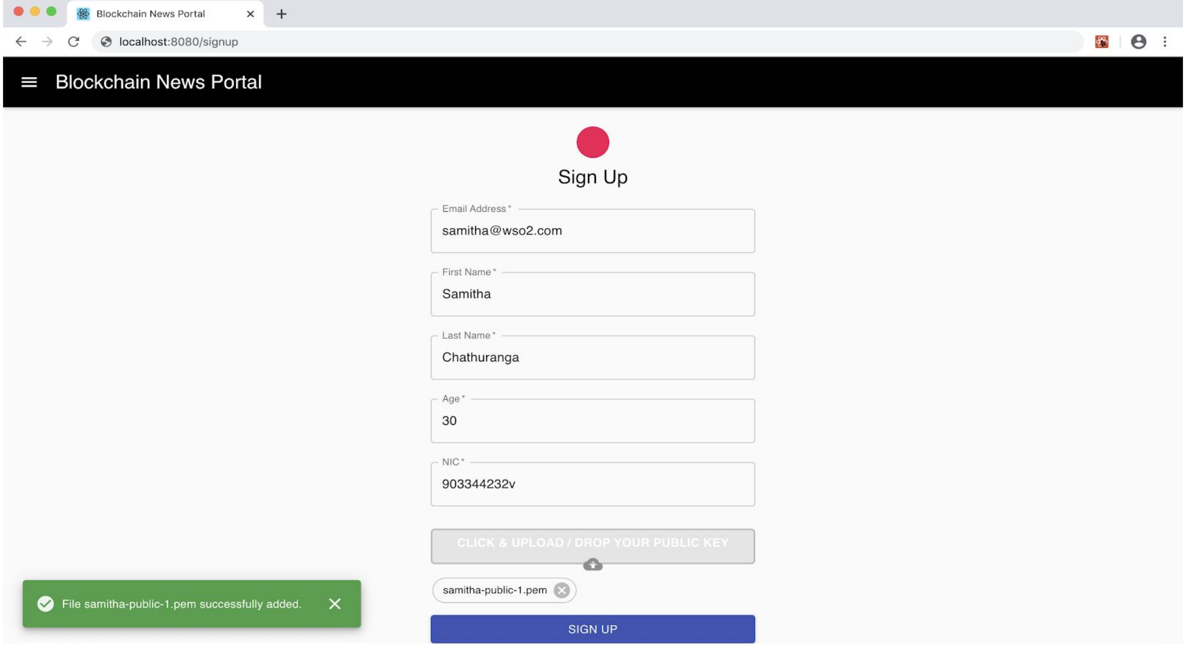
[28] "Top 10 Advantages of Using React.js", *DA-14 Corp*. 2018 [Online]. Available: https://da-14.com/blog/its-high-time-reactjs-ten-reasons-give-it-try. [Accessed: 04- Jan- 2020]

[29] "Single-page application vs. multiple-page application", *Medium*. 2016 [Online]. Available: https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58 [Accessed: 01- May- 2020]

[30] *Material UI*. [Online]. Available: https://material-ui.com/. [Accessed: 21- Nov- 2019]

Appendix A

Some Screenshots of the React Web Application



Email Address * samitha@wso2.com

First Name * Samitha

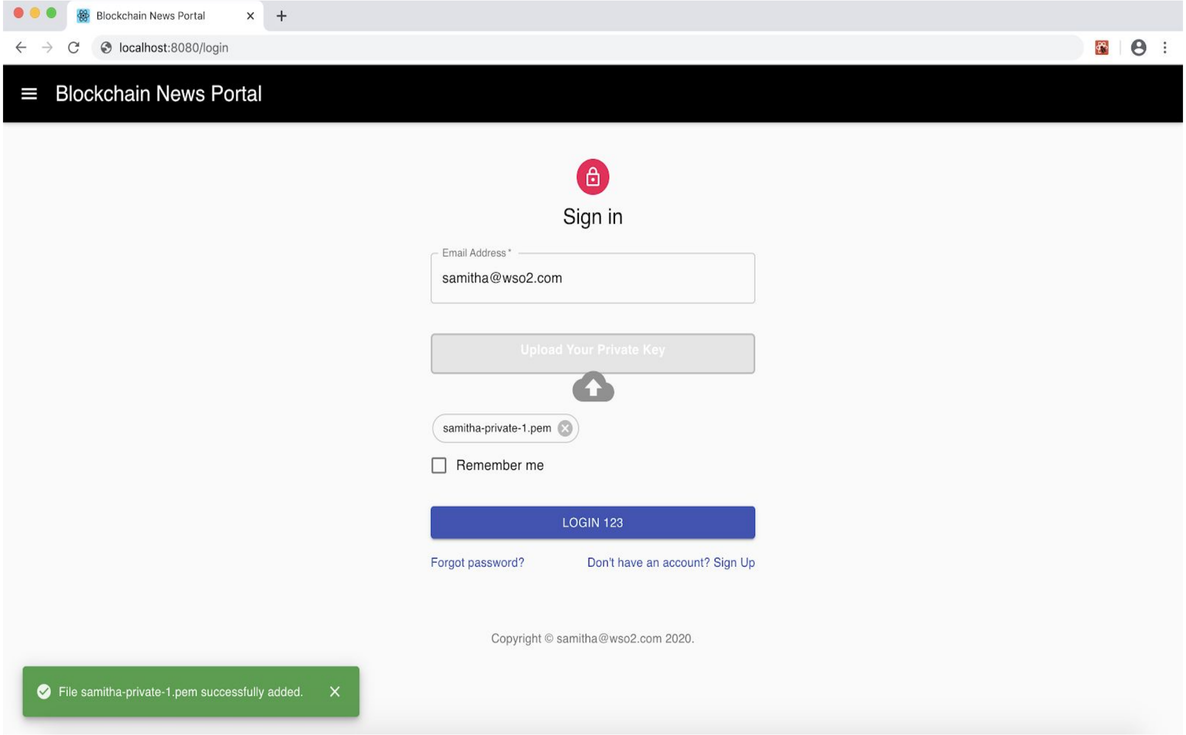
Last Name * Chathuranga

Age * 30

NIC * 903344232v

File samitha-public-1.pem successfully added.

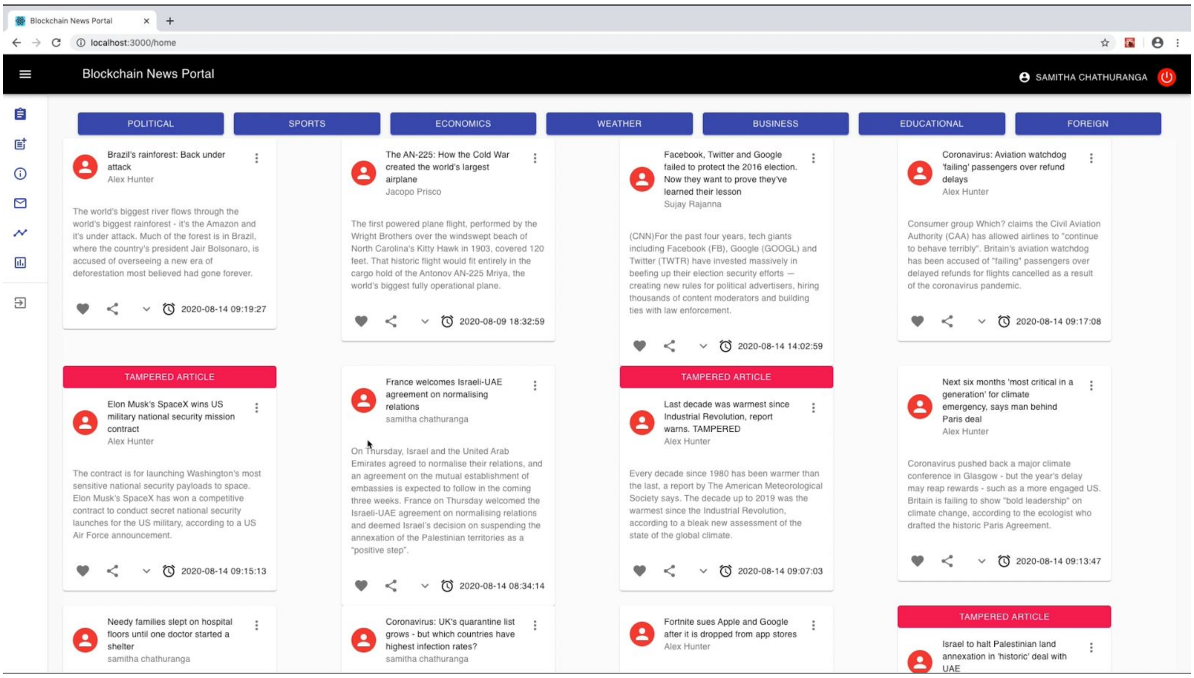
Figure 1: Sign Up Page : After adding information to submit



Email Address * samitha@wso2.com

File samitha-private-1.pem successfully added.

Figure 2: User Login Page: After adding credentials to submit



Brazil's rainforest: Back under attack
Alex Hunter

The world's biggest river flows through the world's biggest rainforest - it's the Amazon and it's under attack. Much of the forest is in Brazil, where the country's president Jair Bolsonaro, is accused of overseeing a new era of deforestation most believed had gone forever.

2020-08-14 09:19:27

TAMPERED ARTICLE

Elon Musk's SpaceX wins US military national security mission contract
Alex Hunter

The contract is for launching Washington's most sensitive national security payloads to space. Elon Musk's SpaceX has won a competitive contract to conduct secret national security launches for the US military, according to a US Air Force announcement.

2020-08-14 09:15:13

Needy families slept on hospital floors until one doctor started a shelter
samitha chathuranga

The AN-225: How the Cold War created the world's largest airplane
Jacopo Prisco

The first powered plane flight, performed by the Wright Brothers over the windswept beach of North Carolina's Kitty Hawk in 1903, covered 120 feet. That historic flight would fit entirely in the cargo hold of the Antonov AN-225 Mriya, the world's biggest fully operational plane.

2020-08-09 18:32:59

France welcomes Israeli-UAE agreement on normalising relations
samitha chathuranga

On Thursday, Israel and the United Arab Emirates agreed to normalise their relations, and an agreement on the mutual establishment of embassies is expected to follow in the coming three weeks. France on Thursday welcomed the Israeli-UAE agreement on normalising relations and deemed Israel's decision on suspending the annexation of the Palestinian territories as a "positive step".

2020-08-14 08:34:14

Coronavirus: UK's quarantine list grows - but which countries have highest infection rates?
samitha chathuranga

Facebook, Twitter and Google failed to protect the 2016 election. Now they want to prove they've learned their lesson
Sujay Rajanna

(CNN)For the past four years, tech giants including Facebook (FB), Google (GOOGL) and Twitter (TWTR) have invested massively in beefing up their election security efforts — creating new rules for political advertisers, hiring thousands of content moderators and building ties with law enforcement.

2020-08-14 14:02:59

TAMPERED ARTICLE

Last decade was warmest since Industrial Revolution, report warns. TAMPERED
Alex Hunter

Every decade since 1980 has been warmer than the last, a report by The American Meteorological Society says. The decade up to 2019 was the warmest since the Industrial Revolution, according to a bleak new assessment of the state of the global climate.

2020-08-14 09:07:03

Fortnite sues Apple and Google after it is dropped from app stores
Alex Hunter

Coronavirus: Aviation watchdog 'failing' passengers over refund delays
Alex Hunter

Consumer group Which? claims the Civil Aviation Authority (CAA) has allowed airlines to "continue to behave terribly". Britain's aviation watchdog has been accused of "failing" passengers over delayed refunds for flights cancelled as a result of the coronavirus pandemic.

2020-08-14 09:17:08

Next six months 'most critical in a generation' for climate emergency, says man behind Paris deal
Alex Hunter

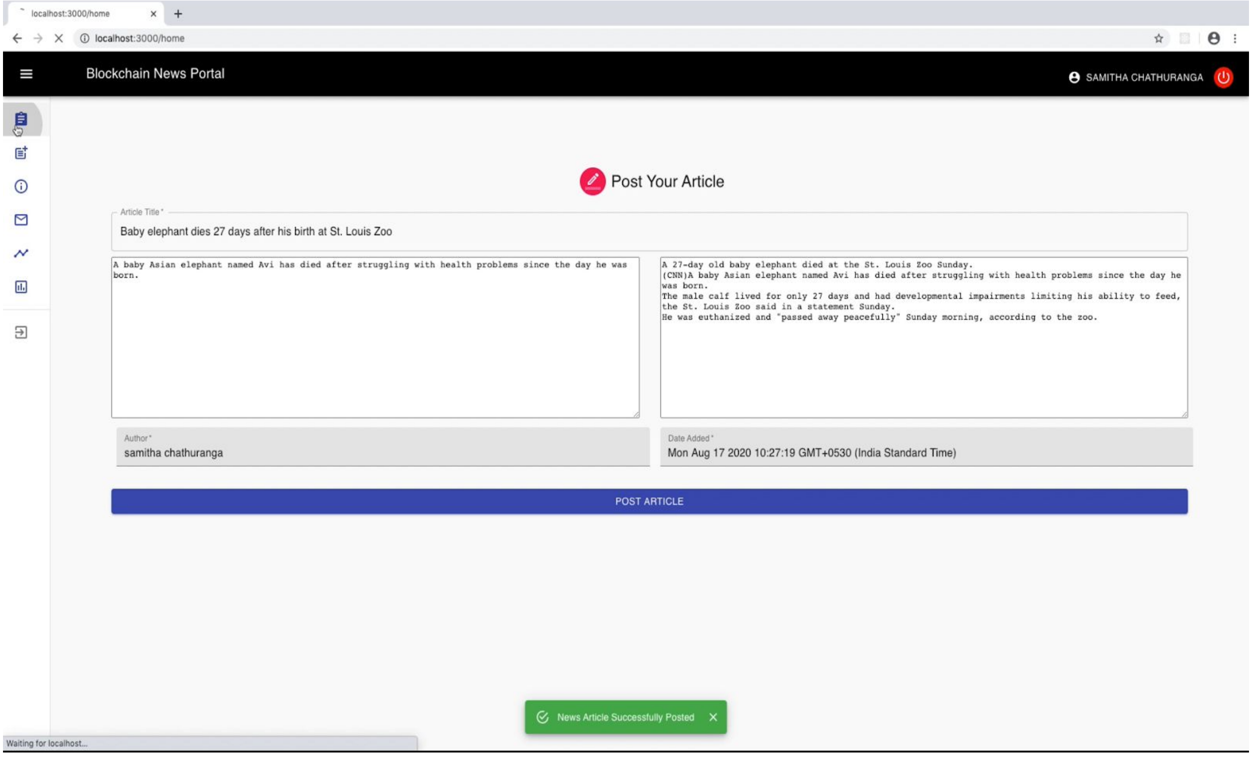
Coronavirus pushed back a major climate conference in Glasgow - but the year's delay may reap rewards - such as a more engaged US. Britain is failing to show "bold leadership" on climate change, according to the ecologist who drafted the historic Paris Agreement.

2020-08-14 09:13:47

TAMPERED ARTICLE

Israel to halt Palestinian land annexation in 'historic' deal with UAE

Figure 3: News Articles Viewing page



Article Title *
Baby elephant dies 27 days after his birth at St. Louis Zoo

A baby Asian elephant named Avi has died after struggling with health problems since the day he was born.

A 27-day old baby elephant died at the St. Louis Zoo Sunday.
(CNN)A baby Asian elephant named Avi has died after struggling with health problems since the day he was born.
The male calf lived for only 27 days and had developmental impairments limiting his ability to feed, the St. Louis Zoo said in a statement Sunday.
He was euthanized and "passed away peacefully" Sunday morning, according to the zoo.

Author *
samitha chathuranga

Date Added *
Mon Aug 17 2020 10:27:19 GMT+0530 (India Standard Time)

News Article Successfully Posted

Figure 4: News Article adding page