A DISTRIBUTED LEDGER SOLUTION FOR MANAGEMENT OF PSYCHOLOGY TEST DATA

A Thesis Submitted to the College of

Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Yalin Chen

PERMISSION TO USE

In presenting this thesis/dissertation in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis/dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis/dissertation work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis/dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

DISCLAIMER

The [name of company/corporation/brand name and website] were exclusively created to meet the thesis and/or exhibition requirements for the degree of Master of Science at the University of Saskatchewan. Reference in this thesis/dissertation to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place Canada
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9
Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

ABSTRACT

Psychology tests are widely used in mental health diagnoses, education assessments, and recruitment assessments. There are several major problems in using the traditional way to manage psychology test data. First, there is no single source of truth for psychology test data due to centralized storage. Second, the data are mutable and have the risk of a single point of failure. Third, people have very weak or no control of their own data. This thesis explores the possibility of adopting the new distributed ledger technology, represented by blockchain, to address the problems. The relevant literature of blockchain and psychology tests was reviewed. A complete academic solution was proposed. It includes a permissioned blockchain, a no-SQL database, a web service, and a front-end. The blockchain stores user profile, metadata of psychology tests, final test scores, and access control data of the tests and test scores. The no-SQL database stores test materials and raw test results. The web service interacts with the blockchain and the no-SQL database. The front-end interacts with the web service. The solution was implemented, and the performance was evaluated. The evaluation results showed that the Post request is slower than the Get request and as the number of clients grows linearly, the latency of the requests grows linearly. The slower latency for the Post request compared to the Get request reflects the time it takes for the blockchain system to write information and change the common ledger status. The solution proposed here provides a new way to manage psychology test data with satisfactory performance. Future research can focus on extending the current solution to other questionnaire data management and to non-questionnaire-based psychology assessment data management.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Figure Number Page Number

CHAPTER 1 INTRODUCTION

Psychology tests are widely used in mental health diagnoses, education assessments, and recruitment assessments. Major problems exist in the traditional psychology test data management using centralized databases. To solve the problems, this thesis explores the possibility of adopting the new distributed ledger technology, represented by blockchain, to psychology test data management.

Distributed ledger is a new decentralized technology in data management and is rapidly gaining popularity. It represents one of the newest and the most exciting trends in our society in moving from a human-scaled society to a computer or machine scaled society. It realizes many new activities that are impossible to realize in a human-scaled society but are possible to realize in a computer scaled society. The benefits it brings can potentially address many critical deficiencies that the traditional centralized systems have.

Although the word test can be interpreted as a set of procedures or a set of questions (i.e., according to Merriam-Webster dictionary, https://www.merriam-webster.com/dictionary/test), this thesis uses the word test to mean a set of questions. Specifically, the word psychology test is used in the following meaning. It refers to a group of specially established questionnaires. Each questionnaire contains a set of carefully designed questions, and the results can be quantified and interpreted in a meaningful way. Sometimes a psychology test is referred to as a scale (e.g., Likert scale, or Wechsler Adult Intelligence Scale, WAIS) or an inventory (e.g., Minnesota Multiphasic Personality Inventory, MMPI).

Three major problems exist in the traditional psychology test data management using centralized databases. They can be solved by applying the distributed ledger technology. First, psychology test data collected from different individuals are traditionally scattered in different databases within different systems. There are major obstacles in establishing a single source of truth for psychology test data using traditional methods. However, the blockchain provides a better way to establish and maintain a single source of truth for psychology test data. It is obvious that a single source of truth for psychology test data would greatly benefit psychology test practices because the data constitute the bases for the practices. Second, psychology test related data are traditionally mutable and have the risk of a single point of failure. The distributed ledger technology makes the data immutable and available everywhere. The data include psychology test data themselves and access control data. If the data are stored on the distributed ledger, then they cannot be altered maliciously after generated, and they are highly available. For psychology test data themselves, using distributed ledger can reduce the risk of faking data and increase the availability of the data. For access control data, using distributed ledger can reduce the risk of unauthorized access to the data and it increases the availability of access control data. This is very important due to the nature of psychology tests. Third, psychology test data are traditionally managed in a centralized way, where people have very weak or no control of their own data. The distributed ledger technology grants people the power to control their own data in a distributed way. In a psychology test, the data collected are usually highly sensitive and are about the subject. People who take the tests ought to have special interests in controlling their own data, and the controlling is required by the Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017).

This thesis proposes a solution that includes a permissioned blockchain, a web service, a no-SQL database, and a front-end. The blockchain stores user profile, metadata of tests, the final score of tests, and access control data of tests and test scores. The blockchain also stores all the transaction logs. The no-SQL database stores test materials and raw test results. The web service directly communicates with the blockchain and no-SQL database. The front-end interacts with the web service. The solution was successfully implemented using a group of techniques. The latency performance was evaluated. It shows that the Post request is slower than the Get request, and as the number of clients increases, the latency increases linearly. Although the solution here focuses on psychology test data management, it also may be of interest to other areas that involve sensitive questionnaire data collection.

The following chapters are organized in the following way. In Chapter 2, the specific research questions will be proposed. In Chapter 3, the relevant literature for the distributed ledger technology represented by blockchain and the relevant literature for psychology tests will be reviewed and summarized. The benefits of using blockchain for psychology test data management will be identified. In Chapter 4, a complete solution will be proposed with a detailed architecture and its implementation. In Chapter 5, the performance of the proposed solution will be evaluated in an experiment and the results will be discussed. In Chapter 6, the solution will be generally discussed and summarized.

CHAPTER 2 PROBLEM DEFINITION

Psychology test data are usually collected and managed as shown in Figure 2.1. First, the test taker takes the test which is administrated by the test administrator. The test materials are acquired from centralized databases. Then data are collected and stored in centralized databases. The database administrator defines access control and grants access to a wide variety of data viewers.

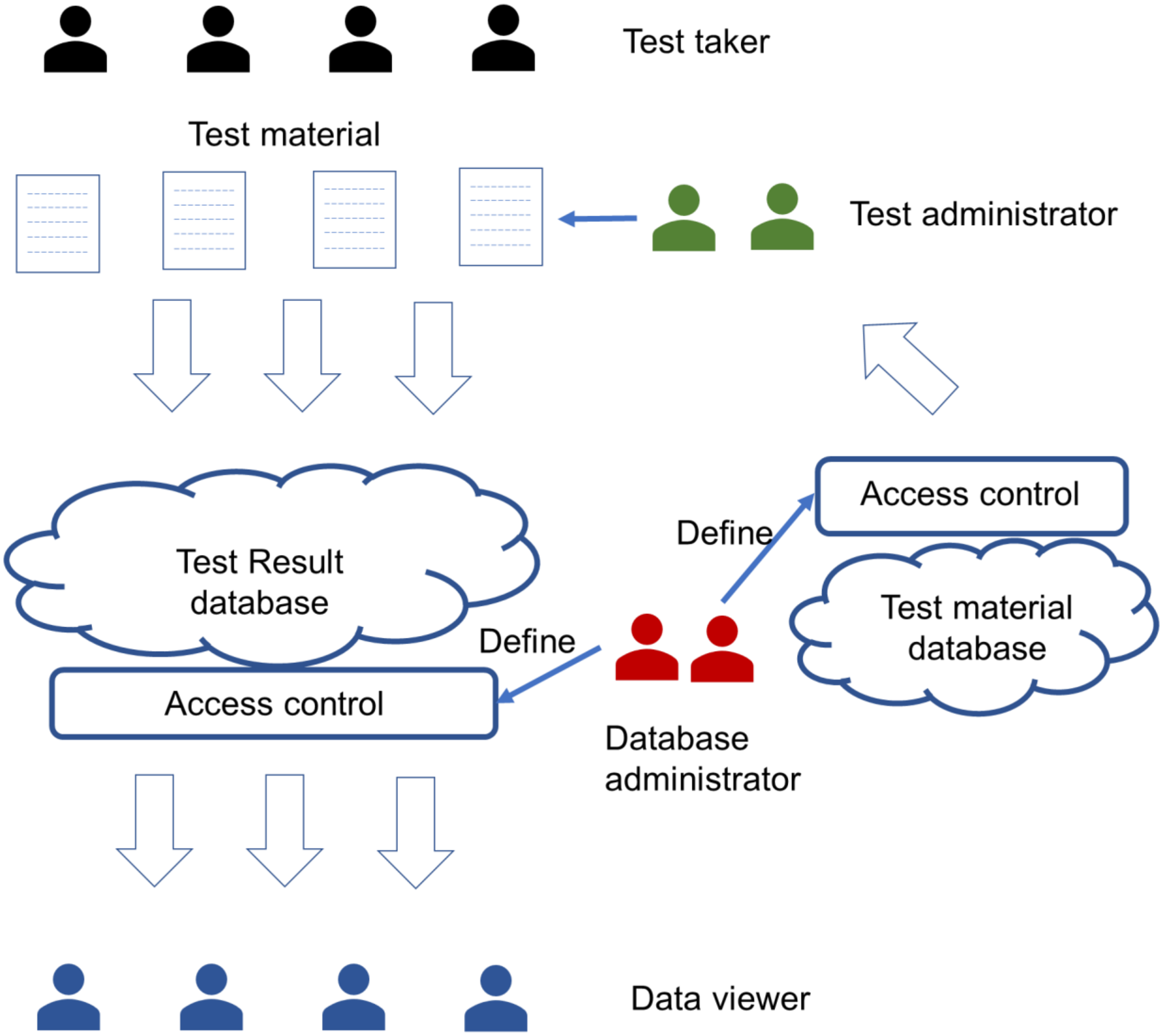


Figure 2. 1. The traditional way to administrate psychology test data

Although there are benefits in using traditional systems for psychology test data management, there are several major problems:

1. There is a lack of a single source of truth for psychology test data. As can be seen from the figure, it is a centralized system. Because that the centralized system requires a central node that ultimately controls everything, and people usually distrust this central node, the test materials acquired from different institutions and test results data collected from different individuals are scattered in different databases within different systems.
2. The data are mutable and with low availability. For instance, the access control can be attacked, causing potential psychology test material data and test result data abuse. This may have serious consequences in certain situations. For instance, if a mentally healthy person can manipulate certain psychology tests that are used in mental health assessments, if the person commits a crime, to reduce or avoid the penalty, the person can manipulate the test results to show that the person has a mental weakness. Moreover, if the central database failed to operate, the data will not be available.
3. People have very weak or no control of their own test result data or test material data. In a psychology test, the test result data collected usually are highly sensitive and are about oneself. People who take the test ought to have special interests in controlling their own test result data, and the controlling is required by the Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017). Traditionally psychology test data are managed in a centralized way. In that way, people have very weak or no control of their own test results data. In theory, it may be possible to allow everyone to define their own access control on their own data through the centralized system. However, the central node has the superpower to override the access control data that the user defined.

How to solve these problems? New technologies in the distributed world may provide an answer. Distributed ledger is a new decentralized technology in data management and is rapidly gaining popularity. It represents one of the newest and the most exciting trends in our society in moving from a human-scaled society to a computer or machine scaled society. It realizes many new activities that are impossible to realize in a human-scaled society but are possible to realize in a computer scaled society. The benefits it brings can potentially address many critical deficiencies that the traditional centralized systems have. Thus, the distributed ledger technology provides us another possibility, with potential benefits, to manage psychology test data. Would it be able to solve the major problems proposed above? This is the major research question for this thesis.

In the following, the research question will be broken down into three specific questions. They include introducing the benefits of using distributed ledger technology in psychology test data management, designing and implementing a theoretical structure of the distributed ledger solution, and evaluating the performance of the solution.

2.1 What are the potential benefits?

The first question is what are the major benefits that a blockchain solution can bring to psychology test data management, that can solve the problems the traditional way faces? To find the answer, relevant literature needs to be reviewed and summarized. They include reviewing the features of the blockchain, reviewing the important aspects of psychology tests, and summarizing the benefits when using blockchain in psychology test data management.

2.2 What would be one solution?

The second question is what one possible architecture would be if one wants to utilize the benefits of blockchain in psychology test data management. The solution intends to be a complete academic solution that includes both a back-end and a front-end. The architecture needs to able to realize the benefits that the blockchain brings to psychology test data management. The architecture needs to contain enough details that make it be able to be implemented.

To propose a solution, there are many questions that need to be answered. For instance, what information should be stored on blockchain? Is there a need for an off-chain database? Which type of blockchain network need to be chosen to build the solution? Who are the basic participants of the network? How do psychology test data to be represented within the network? How do different parties interact with each other within the network? Is a web service needed to communicate with the blockchain? How to register and authenticate users? What functions need to be provided given different types of participants? How the front-end needs to be built? What technologies need to be chosen to implement different parts of the architecture? etc.

2.3 What about system performance?

After the implementation of the solution, the third question is what about the performance of the solution proposed? The performance determines the applicability of the solution. A solution with good performance in a regular environment forms a good foundation for its application beyond academia.

To evaluate the performance, the performance indicator needs to be measured through a carefully designed experiment. Some example questions that need to be answered are, how long does a transaction take? Does the performance significantly decrease as the number of clients increases? etc.

CHAPTER 3 LITERATURE REVIEW

In answering the first research question, which asked for the benefits in using blockchain to manage psychology test data and to provide a theoretical foundation, relevant literature needs to be reviewed and summarized. Chapter 1 reviews the literature on the distributed ledger technology represented by blockchain, including its history, important features, and related applications in one closely related area. Next, this chapter reviews important aspects of psychology tests, including the features of psychology tests, computerized psychology tests, and security and privacy issues. Finally, this chapter summarizes the literature and identifies the potential benefits of using blockchain technology for psychology test data management.

3.1 Overview of the blockchain

3.1.1 The birth of the blockchain

Blockchain was first introduced in 2008 by Satoshi Nakamoto (Nakamoto, 2008). In that paper, a peer-to-peer version of electronic cash system called Bitcoin was introduced. It allows payments to send directly from one party to another party without going through a centralized party (i.e., a financial institution). In the system, each transaction is timestamped, and incorporated into a single ongoing chain. Every node within the network has a copy of the chain. Forming a new record requires proof-of-work, thus changing the record cannot be done without re-do the proof-of-work.

Nakamoto (2008) defines the bitcoin as a chain of digit signatures. That is, the owner transfers the bitcoin to next owner by digitally signing a hash of the previous transaction and the public key of the next owner and adding it to the end of the coin. The payee can verify the signatures. To avoid the double-spend problem, all the transactions need to be published and there need be a single history of the order of the transactions. In this way, the payee can verify whether the payment is the first transaction that the previous owner made for the same coin. If it is, then the transaction is valid. Otherwise, if it is not the first transaction being made for the same coin, then the transaction is invalid because the previous owner has used that coin. In this way, the double-spend problem can be avoided. Thus, a single history of all the transactions that every node agrees on is very critical.

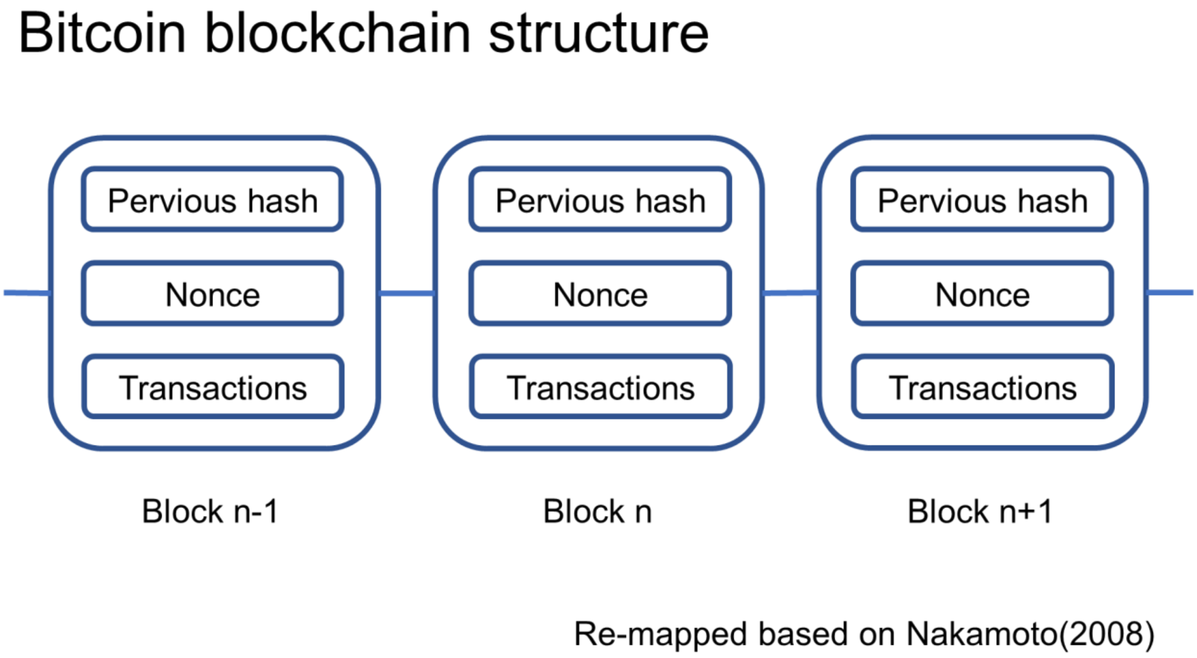


Figure 3. 1. Bitcoin blockchain structure as indicated in Nakamoto (2008)

Bitcoin blockchain proposed a network structure to guarantee a single history of all transactions. Figure 3.1 shows the basic structure of the bitcoin blockchain network. All the blocks are chained together by referring to the hash of the previous block, as indicated by the previous hash section. The proof-of-work is implemented by the nonce component. It involves generating a hash begins with a few zero bits. The transactions are hashed into a Merkle Tree, which is a tree structure of hash values. Because the hardware speed rises, and interest in running nodes varies over time, the difficulty for proof of work changes. It is determined by a varying number targeting an average number of blocks per hour. If the speed is too fast, the difficulty will increase. Otherwise, the difficulty will decrease. Once a block is formed, it can not be modified without re-doing the proof-of-work. If there are other blocks after it, then the block can not be modified unless all the following blocks are re-done. The final decision of the blockchain is represented by the longest chain. It has the greatest proof-of-work involved in it. Thus, if the majority nodes of the whole network are good nodes, the networks will be in a good state. In other words, the attackers need to control the majority nodes of the network to make a valid attack.

The transaction runs in the following ways. First, new transactions are broadcast to all the nodes. Then, each node gets all the transactions and put them into a block. Afterward, each node works on to find a difficult proof-of-work for its block. When a node finds a proof-of-work, it broadcast the block to all the nodes. Then, if all transactions in the block are valid and not already spent, other nodes will accept the block. Finally, nodes express their acceptance of the block by working on creating the next block of the chain, using the hash of the accepted block as the hash to be written in the next block. During the process, if two nodes generated two different versions of the next block simultaneously, the other nodes may receive the two different versions one after another. Then, they based their proof-of-work for next block on the first one they received but save the other branch. If the next block is generated and attached to one of the two tie blocks, then one chain branch will be longer than the other branch. The network was set so that the nodes always consider the longest chain to be the correct one. Thus, the whole network will switch to the long branch.

Because proof-of-work consumes computer powers, there needs to be some incentives to keep the network running. The first transaction of a block starts a new coin owned by the creator of the block. This works as one incentive for the participants of the network. After enough coins are generated, the incentive will switch to transaction fees. Because the total number of bitcoins is pre-set, it is claimed to be inflation-free.

Regarding privacy, the bitcoin blockchain system adopted a different approach than what is used historically. Traditionally, a financial institution (e.g., a bank) works as a central authority and owns all the information. The central authority limits access to the information to their trusted parties. The public usually does not have access to the information. In this way, privacy is protected. In the blockchain system, however, all the transactions are broadcasted to every node within the network. Thus, information is available to everyone. To protect privacy, the bitcoin blockchain system keeps the public keys anonymous. In this way, the public can see that one person has sent some money (i.e., bitcoin) to another person, but whom the person is remains unknown. Additionally, to make it more secure, for each transaction, a new key pair needs to be used. However, as indicated by Nakamoto, the risk is that if the owner of a key is revealed, the transaction can be identified.

Beyond Bitcoin, there are other crypto-currency systems. One popular system is Ethereum. It allows smart contract, which is a computerized contract that can be executed automatically, to run on the blockchain. This grants great power to the utilization of blockchain applications. For instance, stocks and funds can have automatic dividend payments via smart contracts (Tschorsch, & Scheuermann, 2016).

Ethereum allows the user to created applications running on Ethereum Virtual Machine (Ethereum White Paper, 2018). There are two types of accounts on EVM: External owned account and contract account. The external owned account is controlled by a private key and has no code in the account. Users can send a message by creating and signing a transaction. The contract account has code in it. Every time it receives a message, its code will be activated to execute certain functions. Transactions in Ethereum must be paid, and the Ether is the main crypto-currency that is used to pay. Because every transaction has a price, computer power can be saved. Users can participate in the Ethereum mining process to get the tokens that are needed in transactions.

Besides bitcoin blockchain and Ethereum, there are many other crypto-currency systems (Mukhopadhyay et al., 2016; Tschorsch, & Scheuermann, 2016), such as Mastercoin, Ripple, and Peercoin, etc. Because the crypto-currencies are not the focus of the current thesis, we do not introduce the details of these systems.

3.1.2 Blockchain features and consensus algorithms

There is controversy about whether the bitcoin as a crypto-currency has real financial value. After the bitcoin is introduced, people attached real-life currency value to the bitcoin. However, the exchange rate between the bitcoin and the real-life currency changed dramatically during the past years. As the price of bitcoin measured by real-life currency goes up dramatically, many people argue that the rising price of the bitcoin is a bubble and the coin itself have no real-life value at all. However, there is less controversy about the technology, which is named as blockchain, behind the bitcoin. The technology is widely accepted as a valid new technology and it is believed to have many benefits.

Blockchain technology does not necessarily need to be live as a crypto-currency system. People have developed many blockchain applications with functions beyond a crypto-currency system (e.g., Crosby, Nachiappan, Pattanayak, Verma, & Kalyanaraman, 2016; Greenspan, 2015; Tama, Kweka, Park, & Rhee, 2017). For instance, there are blockchain applications in areas such as notary public, music industries, legal institution, decentralized storage, decentralized internet of things (IoT), and so on (Crosby et al., 2016). Because of its potential, the technology is getting more and more popular. For instance, the number of papers published around this topic grows rapidly since 2013 (Yli-Huumo, Ko, Choi, Park, Smolander, 2016).

Regardless of whether crypto-currency is involved or not, blockchain technology has some common structures. It usually contains a common ledger that each node within the network has an exact same copy. The ledger contains multiple chained blocks. Except the first and the last block, each block is connected to a previous block and each block is connected by a subsequent block. In this way, all the blocks form a chain. Each block contains records or so-called "transactions", their timestamps, the current block's cryptographic hash, and previous block's cryptographic hash.

The common structures of the blockchain make it immutable. The immutable refers to that any attempt to modify one block of the blockchain requires to not only modify the block that wants to be modified, but also modify all the subsequent chained blocks, and not only modify the ledger on one node, but also modify the ledger on all the other nodes. More specifically, because any modification of any block will change its cryptographic hash, to chain them together, the immediately subsequent block needs to be modified to contain the modified hash. Consequently, the block after that block needs to be modified to contain the previously modified hash. This process needs to be carried on until the final block is reached. In addition, because every node has an exact copy of the ledger, so after the ledger is modified on one node, all the other copies of the ledger on all the other nodes need to be modified as well. In this way, the blockchain is very hard to be changed after its formation. Note that this only means each established transaction record on the ledger is very hard to be changed. It is not that the final state of the ledger itself cannot be changed. The final state of the ledger can be changed by adding another transaction.

The creation of a blockchain contains the creation of an initial or so-called a genesis block and the creation of all the subsequent blocks. The genesis block is pre-set and does not contain the hash value of the previous block because there is no block before it. After the genesis block is created, all the subsequent blocks are attached to the genesis block one after another, forming a single and unique chain. Certain steps are involved in producing the subsequent blocks. As some authors noted (Christidis & Devetsikiotis, 2016), there are four general steps for the blockchain network to form a new block:

1. The user has a pair of private and public keys. The private key is used to sign the transactions and the public key makes the transaction addressable. The signed transactions are broadcasted to neighbor nodes.

2. The neighbor nodes first verify the transaction according to certain rules and make sure the transaction is valid. Then they broadcast the transactions further. Eventually, the whole network receives the transaction.

3. During a time interval, the transactions collected are ordered and packaged into a timestamped candidate block, a process called mining. Afterward, the mining node broadcast the produced block back to the network.

4. The nodes within the network verify the block and make sure the block contains valid transactions and the hash of the previous block. If true, they add the block into the copy of the ledger they have and apply the transactions contained in the block to update their world view. This marks the end of the formation of one block.

This structure and the formation process of blockchain makes the transactions between parties without trust possible. Each node within the network has a copy of the same common ledger before adding a new block. If each node adds the same new block to their copy of the ledger, then each node will still have the exact same ledger (i.e., reach consensus) after the new block is added. Because each node has the common ledger and it verifies each transaction and each block independently before adding the same new block to their copy of the ledger, nodes that do not trust each other can make transactions without a middle trusted third party.

To agree on which new block will be added or to reach consensus, there needs to be a consensus algorithm agreed by all the nodes. The consensus algorithm guarantees that at a given time, there is one single truth about the history of the records. The consensus algorithm is partly determined by the blockchain network type. The two common types of blockchain are permission-less blockchain and permissioned blockchain. The initial bitcoin blockchain is a permission-less blockchain, which means that everyone can join. It is also called public blockchain. As the blockchain technology develops, another type of blockchain was developed. It is called permissioned blockchain. In this type of blockchain, only trusted parties can join. The permissioned blockchain is also called private blockchain. In permission-less or public blockchain, the consensus algorithms need to have special constraints to avoid malicious behaviours because users can be anyone. For permissioned or private blockchains (e.g., Hyperledger Fabric), because only interested and trusted parties can join, this eliminates a lot of potential malicious behaviours and it puts fewer limits on consensus algorithms that can be used. Also, in permissioned blockchain, because the parties are all interested in joining the network, and there is a purpose for building the network, there is no need to offer extra incentives for the miners to generate new blocks.

The bitcoin blockchain uses a mechanism called proof-of-work (PoW), as it has been described above. The PoW requires the node to perform a task that is hard to do but easy to be verified. The first node which finished the required work get the privilege to write a new block into the blockchain (occasionally as we mentioned before, there may be a fork, where two nodes generate the proof of work at the same time). The PoW limits the amounts of malicious behaviour because every PoW consumes computer power. Beside proof-of-work, there are other consensus algorithms such as proof of stake (PoS), Practical Byzantine Fault Tolerance (PBFT), etc.

The PoS algorithm is an algorithm that requires fewer computer power relative to PoW in ordering and generating new blocks. PoS requires a crypto-currency. The chance for a node to mine a new block (i.e., order the transactions and add a new block) is related to the balance that the node has (Christidis & Devetsikiotis, 2016). The implementation of PoS is relatively complicated and it is based on coin age (Tschorsch, & Scheuermann, 2016). The coin age refers to the amount of the coins times the period of the holding. For instance, if one person has an x number of coin and owns the coin for y days, then the coinage is x*y. However, when one spends the coins, the coin age is destroyed. In PoW, the longest chain is deemed as the single truth of the recorded history, but in PoS, it takes the chain with the highest total sum of the destroyed coin age. In this way, if one wants to attack the blockchain, one needs to own a large number of coins. However, the owning of a large number of coins decreases one’s motivation to attack the blockchain (Tschorsch, & Scheuermann, 2016).

Practical Byzantine Fault Tolerance (PBFT) is an algorithm used in the permissioned blockchain (Christidis & Devetsikiotis, 2016). PBFT is proposed to solve the Byzantine Generals Problem (Castro, Liskov, 1999). It involves a leader node and some other nodes. If the leader node crashes or if it exhibits arbitrary behavior, the leadership can be changed by the rest of the network via a voting mechanism. PBFT works if less than one-third nodes are at fault. There are also a few variants of PBFT algorithms adopted in blockchain applications (Christidis & Devetsikiotis, 2016).

The blockchain system is not without any disadvantage. The two most obvious disadvantages are 51% attack and storage-speed trade-off. 51% attack refers to that attacking more than half of the nodes in the network. Bitcoin blockchain requires that the majority of the nodes (i.e., 51%) be good nodes to avoid malicious attacks. Thus, if 51% of the nodes are attacked, the blockchain may not work as what it intends to. For permissioned blockchain, if the blockchain adopts a Practical Byzantine Fault Tolerance algorithm, then it requires less than one-third nodes to be attacked to function properly. It may seem that the permissioned blockchain is easier to be attacked. However, because each node needs to be recognized and trusted before joining the network, the likelihood for it to be an attacker is small. The storage-speed trade-off refers to that the more information stored on the blockchain, the slower the blockchain operations will be. There are reasons for that. First, within the blockchain network, the ledger contains all the transactions. Thus, the more the information, the larger the ledger will be. Therefore, it requires more time or more resources to handle the ledger related process (e.g., query, storage, etc.), and it takes longer for the new node to join as the blockchain grows. Second, because every transaction is broadcasted and verified by every node within a blockchain network, thus the more information each transaction contains, the slower the process will be. Consequently, blockchain is usually used to store only a small amount of necessary information.

3.1.3 Blockchain applications in medical data management

To the best of my knowledge, currently, there is no literature involving the application of blockchain technology in psychology test data management[1]. As a result, this section reviews a few representative studies about the blockchain applications in one closely related area, the medical record management area. The medical record management area is chosen because although there are major differences between psychology test data and medical data, they are like each other in many aspects. For instance, both are about oneself: medical data are about the physical status of oneself whereas psychology test data are about the mental status of oneself; both require a similar level of privacy protection; both are operating on a centralized system traditionally; and in both areas, patients or clients have a natural need in controlling their own data. Thus, looking into blockchain applications in medical data management area would provide insights in developing a blockchain solution for psychology test data management. Recently, a few

[1] During the development of this thesis, I have learned that a blockchain solution was developed for mental health data management called WellLinc (retrieved on July 24, 2018 from https://vitalhub.com/welllinc-blockchain-solution/). They claimed that they joined the Hyperledger to build a project for mental health data management on January 30, 2018 (retrieved on July 24, 2018 from https://globenewswire.com/news-release/2018/01/30/1314154/0/en/VitalHub-Joins-Hyperledger.html), which is after the starting date of this thesis project. However, there is very limited information that is available publicly about the project.

studies concerning the application of blockchain in medical record management area have been conducted.

Azaria, Ekblaw, Vieira, and Lippman (2016) proposed a MedRec system for electronic medical records management using blockchain technology. They argued that they are the first to introduce a fully functional prototype that applies blockchain technology to medical records management. They built their solution based on Kish and Topol (2015) and Zyskind, Nathan, and Pentland (2015). The solution is built upon Ethereum blockchain. In the solution, the blockchain stores permissions that regard data ownership and viewership. It logs patient-provider relationships that associate with a medical record. It also contains pointers for execution on external databases and hash values of the medical record.

To implement the solution, the authors designed three types of contracts. The first one is called Registrar Contract (RC). The RC maps participants identification information (i.e., strings) to their Ethereum address identity (equivalent to a public key). At that address, there is a Summary Contract (SC), which will be explained later. The second smart contract is called the Patient-Provider Relationship Contract (PPR). PPR is issued when one node (e.g., provider) stores and manages medical record for the other (e.g., patient). It defines a collection of data pointers and their access permissions linked to the records stored in the provider. Each data pointer contains a SQL query which executed on the off-chain database and returns a query result. The SQL query is affixed with the hash of the data subset to ensure that the data are not changed. Other information stored includes the hostname and port of the database. The data and queries are maintained by the providers when new records are added. A dictionary maps viewers' addresses to a list of additional query strings. The patient can decide what to add to the viewers' list to control who has access to what portion of the data. A third contract is called Summary Contract (SC). SC holds a list of references to PPRs, which contains all the history of the patient's medical records. It also enables user notifications, when a new patient-provider relationship is established. The users can choose whether to accept, reject, or delete relationships.

Kuo, Kim, Ohno-Machado (2017) systematically reviewed the application of the blockchain technology in biomedical and health care area. The authors summarized that as one important application area of the blockchain, medical blockchain applications usually are used to store health-related data for sharing, exchanging, analyzing, recording and validating purposes. The most discussed applications can be categorized based on their main goals to exploit the blockchain-stored data. These categories were labeled as improved medical record management, enhanced insurance claim process, accelerated clinical/biomedical research sections, and advanced biomedical/health care data ledger. In the current context, the most relevant category is improved medical record management.

The authors summarized the benefits and the use cases of adopting blockchain in improving medical data management. First, it offers decentralized management, where the use case is patient-managed health care records. It removes the obstacles for patients acquiring and sharing their data. Second, it offers an immutable audit trail, where the use case is unalterable patient records. Third, it offers data provenance, where the use case is source-verifiable medical records. The data are signed by the source. Fourth, it offers robustness/availability, where the use case is the reduced risk of patient recordkeeping. It can avoid centralized hack where many patients' records may leak at one time. Fifth, it offers security/privacy, where the use case is increased safety of medical records (see also Angeletti, Chatzigiannakis, & Vitaletti, 2017). Data can be encrypted using the patient's public key and thus can only be decrypted with the patient's private key. In this way even data are leaked, there is no practical way to decode the patient's data. Regards to the application themselves, the authors mentioned Healthcare Data Gateways, MedVault, Fatcom, BitHealth, Gem Health Network, and others, but no detail of these applications was provided.

Other studies have examined the application of blockchain in medical record management. For instance, one paper developed a blockchain solution that can apply to cancer patients' data management (Dubovitskaya, Xu, Ryu, Schumacher, & Wang, 2017). They employed permissioned blockchain to maintain the metadata and the access control and a cloud service to store encrypted patient's data. Although, store all the data on blockchain may be perceived as infeasible and have poor performance (Rifi, Rachkidi, Agoulmine, & Taher, 2017), some papers proposed on-chain storage of all the data. For instance, a paper about healthcare data gateways proposed a blockchain storage of all patients' data, but it treated decentralized blockchain as a centralized storage to storage all patients' data and focused on the design of the gateways to interact with the blockchain (Yue, Wang, Jin, Li, & Jiang, 2016). Another paper also proposed on-chain storage for medical data, but the description is not clear enough (Liu, Zhu, Mundie, & Krieger, 2017). There are also some studies focus on part of data management in health care (e.g., data sharing between different data centers) (Xia et al., 2017; Zhang, Xue, & Huang, 2016; Zhao, Zhang, Peng, & Xu, 2017). There are other applications proposed (Baxendale, 2016, some were also mentioned in Kuo, Kim, Ohno-Machado, 2017), such as Factom (Wiese, 2016), MedVault (Nichol, 2016), Gem Health Network (Prisco, 2016), healthbank (Mettler, 2016), BitHealth (Sarwal & Insom, 2018), and others (Beninger, & Ibara, 2016). But they lack publicly available summary of technical details.

3.2 Overview of psychology tests

Several common words are used in describing a psychology test, including survey, questionnaire, and psychology test. They are related to each other and sometimes they are used interchangeably.

Survey refers to ask people questions to acquire information. It refers to a research process including design, method and stimuli construction, collecting data, and analyzing data, etc. Questionnaire refers to a set of problems that is used as a tool to acquire information. Most survey research relies on the use of questionnaires as instruments to measure variables (Shaughnessy, Zechmeister, & Zechmeister, 2006). The variables can be many things. For instance, one common variable is a demographic variable, which describes the characteristic of people who are surveyed. Another variable could be people’s preferences or attitudes, which can be measured by self-reported scales (e.g., to rate an item on a 5-point scale, where 1 represents strongly disagree and 5 represents strongly agree). For self-reported measurements, a well-constructed questionnaire need have good reliability, which refers to the consistency of measurement, and validity, which refers to the truthfulness of a measure. Survey and questionnaire are relatively broad concepts and their usage is not limited to psychology.

Psychology test is a more specific word that is used psychology. The word psychology test is a bit confusing, partly because as a noun word, the test could mean a set of procedures/activities or a set of questions (i.e., according to Merriam-Webster dictionary, https://www.merriam-webster.com/dictionary/test). For test as a set of procedures, psychology test could refer to the whole process to run a test. For test as a set of questions, a test is like a questionnaire. In this meaning, psychology test refers to specially established questionnaires each of which contains a set of carefully designed questions and the results can be quantified and interpreted in a meaningful way. In this way, the test shares a similar meaning as scale (e.g., Likert scale, or Wechsler Adult Intelligence Scale, WAIS) or inventory (e.g., Minnesota Multiphasic Personality Inventory, MMPI). If not specifically described, the current thesis uses the word psychology test in the later meaning.

Psychology tests are used to assess ability, personality, and behaviour. They can be used in the selection process for job interviews, education assessments, and mental health diagnoses (The British Psychological Society, 2018). There is a long history of using psychology tests (Sackett, Lievens, Van Iddekinge, & Kuncel, 2017). Generally, psychology tests can be classified into two categories. One is called the measurement of typical performance. It includes those designed to assess personal qualities, such as personality, beliefs, learning styles, and interests; abnormal phenomena such as anxiety, depression, attention deficit hyperactivity disorder (ADHD), etc., and to measure motivation or “drive”. These tests usually have a time limit and there are no right or wrong answers. The other category is called measures of maximum performance. It includes those designed to measure performance, such as tests of ability, aptitude or attainments. These tests either have right or wrong answers or have tasks that can be performed better or worse (The British Psychological Society, 2018).

3.2.1 Important features of psychology tests

Besides the specific questions included in a psychology test, there are other features it encompasses: reliability, validity, and interpretation.

Reliability refers to a test’s degree of stability, consistency, predictability, and accuracy (Groth-Marnat, 2002a). For instance, the test re-test reliability measures to what extent the same result can be acquired if a person is tested again on the same test, compared to the initial test. If the test results are consistent with each other, then the reliability is high, otherwise, the reliability is low.

Validity is another crucial issue in test construction. It assesses what the test is to be accurate about. In other words, it measures to what extent it measures what it intends to measure. A test can be reliable but not valid. For example, a test measures a person’s musical preference might erroneously state that it is a test of creativity. The test might be reliable in the sense of that if it is given to the same person, it produces similar test re-test results. However, it would not be valid in the sense that it does not correlate with other similar measurements of creativity. Although a test can be reliable without being valid, a test cannot be valid but not reliable (Groth-Marnat, 2002a). That is, a valid test must have an adequate level of reliability.

Interpretation refers to how to interpret the test results. A typical psychology test is a quantified test where a raw score can be acquired. However, in some cases, this raw score needs to be converted into a standard form to facilitate the interpretation (Groth-Marnat, 2002a; The British Psychological Society, 2018). Some tests have norms that reflect the distribution of the scores by a standardization sample drawn from a certain population. Norms provide information about the distribution of scores. Scores can be converted into numbers that show how a person has performed relative to this population by comparing it with the norm means and standard deviations. Norms are important because they determine how to interpret the test results. For instance, if a score from a person is converted to a standard score, we can say a person's performance is equivalent to a certain percent of the people in a certain population. In addition, a much more powerful approach is to use the relationship between test scores and external measures of interest, such as educational outcome, job success, categories or mental dysfunction, etc. For example, if validation studies show that a person who scored less than 10 on a test has a 50 percent failure rate in a training course, who scored between 10 to 15 have a 35 percent failure rate and who scored 16 above have a 20 percent failure rate, then the people can be classified into different categories of predicted failure rate based on their test results (The British Psychological Society, 2018).

3.2.2 Computerized psychology tests and testing

The utilization of computerized psychology tests and testing has grown exponentially. In the second half of the last century, computers were first utilized in administration and scoring personality questionnaires and cognitive tests (Butcher, Perry, & Hahn, 2004). By 1999, 40% of psychologists stating that they used some form of computer-assisted testing (McMinn, Buchanan, Ellens, & Ryan, 1999). Groth-Marnat (2002b) concluded that the future of psychological assessment will probably be largely influenced by the trends toward computerized assessment.

There are several benefits of computerizing psychology tests or testing. Because the computer can present the items the same way each time, with the possibility to randomize the order (Dede, Zalonis, Gatzonis, & Sakas, 2015), it can improve the reliability, standardization, and objectivity of the testing activity (Schulenberg, & Yutrzenka, 2004). Through running the testing with the computerized test, the scoring can be automatic and there will be fewer errors, the report can be generated automatically, and the data can be stored electronically and be reviewed remotely (Groth-Marnat, 2002b). The time efficiency of testing using a computerized test will reduce the cost (Groth-Marnat, 1999; Groth-Marnat & Edkins, 1996) and enable psychologists to focus on other parts such as intervention strategies. Testing using a computerized test also allows collecting other types of data, such as response latency (Schulenberg, & Yutrzenka, 2004) or error rate (Dede, et al., 2015). They can also potentially be combined with other innovative assessment and intervention options that are unavailable with traditional measures (Bilder, 2011; Schultheis & Rizzo, 2008). For instance, they can be combined with brain imaging technology (Roalf et al., 2014) and provide more accurate and detailed information about the patients. Moreover, it enables the assessment of larger groups of people, provides automated development of databases that permit direct comparison between different groups, and enable the power of the new form of test which combines visual and auditory stimuli (Dede et al., 2015).

There are already many computer software applications that computerized certain psychology tests and the testing process. For instance, in 2012, Pearson Canada Assessment, a group who has published many important psychology tests, released an iPad-mediated cognitive test administration (Vrana, & Vrana, 2017). This system is available for several cognitive tests published by Pearson and is repaid developing (Noland, 2017). Besides this system, there are also Q-global, Q Local Scoring and Reporting Software, and PsychCorpCenter-based Scoring Software, etc (Pearson, 2018). They computerized psychology tests and enhanced the efficiency of the whole testing process. Besides Pearson Canada Assessment, there are many other systems available (e.g., Bauer et al., 2012).

Although more and more psychologists started to use computerized tools and more and more computerized tools are developed through the years, the number of applications is disproportionally low in respect to the new method’s potential (Witt et al., 2013). For instance, regards to the development stage of computerized testing in neuropsychology, Rabin et al. (2014) investigated the utilization rates of computerized tests and test batteries (i.e., a group of tests) among clinical neuropsychologists in the United States and Canada. With 512 valid responds, they concluded that although computerized testing is a highly active area with new computerized tests continuously emerging, there is still a lot of room left for adopting computerized testing in neuropsychological practice. There may be various reasons why the development of computerized tests and testing is still in its early stage. One reason might be that there are too many versions of the computerized test. There are single tests translated from traditional ones and new ones with only a computerized version. Some computerized tests are entirely independent, whereas some depend on the examiner. Some are designed for specific purposes whereas some are designed for general purposes. A second reason might be that the computerized test or testing may limit the patients to ones who are familiar with computers. A third reason might be that the validity and reliability of the computerized test may be different compared to the traditional way (Dede, et al., 2015). Most importantly, another reason might be that some psychologists are lack of familiarity with computerized tests. For instance, Rabin et al. (2014) showed that the use of computerized tools is more frequent among younger and newer practitioners. Generally, younger and newer practitioners are more familiar with computerized tools. This might indicate that computerized tests will be more and more popular as younger practitioners enter the field and people are more familiar with computerized testing.

3.2.3 Security and Privacy

Given the special nature of psychology tests, there are special requirements for security and privacy. Psychologists are required to protect the test materials as well as the test result data[2] (American Psychology Association, 2017). The test materials refer to manuals, instruments, protocols, and test questions or stimuli and do not include test result data (American Psychology Association, 2017). Psychology tests are different from physical status tests in important ways. For instance, a person can learn a psychology test material and respond to it accordingly. A good psychology test usually requires years of effort to develop, but it can be learned relatively quickly. If psychology test materials are widely available, it would be relatively easy for people to learn the test, including the results interpretation. Thus, to acquire certain interpretations, one can manipulate the test results by responding to the test in a certain way. That would be a serious problem for psychology tests because, in that case, the test is not able to measure what it intends to measure. This may have serious consequences in certain situations. For instance, if a mentally healthy person can manipulate certain psychology tests that are used in mental health assessment, if the person commits a crime, to reduce or avoid the penalty, the person can manipulate the test results to show that the person has a mental weakness. In addition, in providing psychology assessment or scoring services, the service needs to accurately describe the purpose, norms,

[2] Note. Here the term test results data represents the term test data in the Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017). This thesis uses the term test data in a more board way, where test materials are also a type of psychology test data.

validity, reliability, and applications of the procedures and any special qualifications applicable to their use (American Psychology Association, 2017). These data need to be protected as well. The test result data refers to raw and scaled scores, client/patient responses to test questions or stimuli, and psychologists' notes and recordings concerning client/patient statements and behavior during an examination. They need to be well protected will access restrictions. Except for extreme conditions, such as people will be harmed, data will be misused or misrepresented, or required by law or court orders, psychologists can only release the test result data according to a client/patient request (American Psychology Association, 2017).

Traditionally, it is recommended that all tests should be kept locked in a secure place and no untrained person should be allowed to review them. One should not duplicate any copyrighted material. The raw data should not be ordinarily released to persons who may misinterpret them. The clients should have the right to access their own data. Also, they also have the right to release their own data to another person they designate, and such requests should be made in writing (Groth-Marnat, 2002b; Zuckerman, 1997).

The Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017) requires psychologists to carry certain duties to protect confidential information and patients' or clients' privacy. It states that psychologists have primary obligations to take reasonable precautions to protect confidential information obtained through or stored in any medium. Psychologists are required to inform clients/patients of the risks to privacy and limits of confidentiality if they offer services, products, or information via electronic transmission. Psychologists discuss confidential information only with persons clearly concerned with such matters. In disclose confidential information, there should be appropriate consent of the organizational client, the individual client/patient, or another legally authorized person on behalf of the client/patient unless prohibited by law. If without the consent of the individual, psychologists can only disclose confidential information as mandated by law, or where permitted by law for a valid purpose (i.e., provide needed services, obtain professional consultations, protect someone from harm, or obtain payment for services in which instance disclosure is limited to the minimum that is necessary).

In practice, however, the guidelines are hard to achieve according to various reasons (Groth-Marnat, 2002b). For instance, medical practices require most of the patient information (including psychological test results) stored together, and all the members of the treatment team have access to the information. This may reflect a conflict between the benefit to the patient and patient privacy. For one thing, if all treatment teams having access to the patient's records, there may be a better treatment plan. But the patient does not have or have very little control over who and where information should go. In addition, security of test results can also be affected by the fact that other organizations (e.g., insurance company) want to access patient records. This is generally a big issue in the general health care domain. Patient records are stored in centralized databases, and many different parties have access to the databases (McMinn et al., 1999; McMinn, Ellens, & Soref, 1999). In this way, a central party controls the data, and it shares data with different other institutions without the participation of the patients.

In some cases, security and privacy concerns are preventing the adoption of a computerized test in clinical practices (Rabin et al., 2014). Indeed, researchers have argued that, with the increase of the computer use in psychology testing, clinicians may unknowing violate ethical standards relating to confidentiality if access to computer test results were not protected (Schulenberg, & Yutrzenka, 2004). For instance, put confidential information on the computer hard drive may compromise confidentiality in certain situations (e.g., computer repair) (Schulenberg, & Yutrzenka, 2004). In recommendations, Schulenberg and Yutrzenka (2004) suggested that computers containing confidential information (e.g., test results, reports) should be stored in a secure area and access to clinical files should be restricted to persons who have an access code. Instead of using client names, using numbers can minimize the presence of identifying information. Generally, the computerized test user or service provider is responsible for the security of computer-generated materials (Butcher, 2003).

3.3 Summary and the blockchain solution for psychology test data management

This chapter first reviewed the creation of blockchain technology and the initial bitcoin blockchain network. Then, the common features of blockchain were described and the blockchain applications in medical record data management area were reviewed. Afterward, the important features of psychology tests were introduced, the development of computerized psychology tests or testing were described, and the security and privacy issues were discussed. Next, this chapter briefly summarized them in the following and explored the potential benefits of using blockchain for psychology test data management.

Blockchain technology was first introduced by Satoshi Nakamoto (Nakamoto, 2008) with a crypto-currency called bitcoin. It removed the need of a central party in financial transactions but guarantees the security and successfulness of the transactions. The transactions were done by a computer network. Within the network, each node verifies each transaction and keep a copy of a ledger which represents a single truth of history. The process was done through a proof-of-work consensus algorithm. Soon after bitcoin was introduced, other crypto-currencies were developed, such as Ethereum. However, crypto-currency is not necessary to utilize the benefits that blockchain brings. Many other blockchain applications without crypto-currencies were also developed.

There are some common features for blockchain networks. It usually contains a common ledger that each node within the network has the exact same copy. The ledger contains multiple chained blocks. Except the first and the last block, each block is connected to a previous block and each block is connected by a subsequent block. Each block stores logs of the event (i.e., transactions). This structure makes blockchain immutable. There are permission-less blockchain networks and permissioned blockchain networks. To agree on which new block will be added or to reach consensus, there needs to be a consensus algorithm agreed by all the nodes. The consensus algorithm guarantees that at a given time, there is one single truth about the history of the records. Some examples of these algorithms are proof-of-work, proof-of-stake, and Practical Byzantine Fault Tolerance (PBFT), etc. The blockchain system is not without any disadvantage. The two most obvious ones are 51% attack and storage-speed trade-off.

There are already some blockchain applications in medical data management, which potentially provide some insights in developing a blockchain solution for psychology test data management. In medical data management area, there are proposed blockchain networks that stores only a part of medical records, such as access control, and there are also proposed blockchain networks stores all the medical data on a blockchain. There are benefits in utilizing blockchain to manage medical records, such as data are decentralized, immutable, highly available, and secure, etc.

The term psychology test is used to represent specially established questionnaires. Psychology tests are used to assess ability, personality, and behaviour of human. A typical example of a psychology test would be an IQ test. They can be used in the selection process for job interviews, education assessments, and mental health diagnosis. Typically, there are three most important features people are looking for when using a test, those are reliability, validity, and interpretation. The reliability of a test refers to the consistency of the test results when it is administrated multiple times. The validity measures to what extent it measures what it intends to measure. The interpretation of the test results refers to the meanings of the test results.

The computerization of psychology tests and testing is growing exponentially. There are several benefits in using computerized psychology tests and testing, such as better controlling of the presentation, automatic scoring, and reporting, remote access of the result, less cost timely and financially, enables new ways of testing, etc. There are already many computerized psychology tests or testing software, but overall the computerization of psychology tests and testing is still in its early phase.

Psychology tests have special characteristics. Thus, there are special requirements for security and privacy protection. The Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017) requires psychologists to carry certain duties. For instance, the test materials need to be well protected to prevent the abuse usage, the test results data need to be protected for patients' or clients' privacy. In disclose confidential information, there should be appropriate consent of the organizational client, the individual client/patient, or another legally authorized person on behalf of the client/patient unless prohibited by law. In practice, however, the guidelines are hard to achieve. Indeed, security and privacy concerns are preventing the adoption of computerized tests in practices. One major concern is caused by the centralized storage of information. The information is usually stored in a centralized database where many different parties have access to the databases. In this way, the database administrators can grant access where patients or clients do not have or have very little control over it. Also, a large amount of information may be leaked through a centralized attack. For instance, in attacking the centralized access control to get access, a larger amount of the test material and test results may be leaked at once and the test results may be altered.

Based on the literature review, we can see natural needs and benefits that blockchain technology can bring to psychology test data management. On one hand, blockchain technology is developing and gaining popularity rapidly. It represents the newest trend in the transition from a human-scaled society to a computer-scaled society. It removes the needs for centralized authority, thus, it potentially can apply to and brings radical change to an area that currently relies heavily on a centralized authority. The area of psychology tests is such an area. On the other hand, within the psychology test area, utilizing computerized psychology tests and testing is an important and rapidly growing trend due to the benefits it brings. Thus, developing computerized psychology test systems are natural needs in the psychology test area. Third, the special nature of psychology tests has special requirements in psychology test data management. The current centralized system is not able to meet the requirements nicely and sometimes the centralized data management system conflict the requirements. Utilizing blockchain technology can potentially address these problems nicely in a different way.

Before going any further, it is important to identify the specific benefits that the blockchain technology can bring to psychology test data management. Through the literature reviews above, three major benefits can be identified in applying blockchain technology to psychology test data management. These benefits are as the followings.

First, the distributed ledger technology provides a better way to establish and maintain a single source of truth for psychology test data. The goal of psychology tests is to provide accurate information about a person. Generally, the more information at hand, the better the result would be. Thus, it is important to gather information easily. The ideal way to provide all psychology test results of a person is to retrieve all the information from a single source of truth, which also includes all the test materials. It is true that by using traditional techniques such as centralized databases, it is possible to provide a single source of truth. For instance, if everyone uses one single centralized database, then the single source of truth can be established. However, traditionally, the test materials acquired from different institutions and test results data collected from different individuals are scattered in different databases within different systems. Part of the reason is that the centralized system requires a central node that ultimately controls everything. All the other non-central nodes in the system have less control over the information. Thus, all the other nodes need to fully trust the central node if everyone is using the same centralized database. However, the full trust of the central node is very hard to be established. In a blockchain network, there is no central node and every node is equal. Every node has a copy of the common ledger and independently verifies any change in the ledger. Also, all the verified changes that have been made are immutable. In this way, the full trust of a central node is not needed in establishing a single source of truth. Thus, it is more possible to establish and maintain a single source of truth for psychology test data using blockchain technology.

Second, the distributed ledger technology makes psychology test related data immutable and available everywhere. The data include psychology test data and access control data. Because the blockchain makes things stored on it immutable, it can prevent part of potential psychology test materials and test result data abuse. For instance, because of the immutable records on the blockchain, a person who has taken a specific test through blockchain can not claim that he has not done it and do it again to get a score he wants. Also, the test result data can not be changed; this reduces the risk of faking test results. In addition, the nature of psychology tests requires the access control information be secure and has high availability. In the traditional centralized systems, however, the access control of data is available in a central place, make it vulnerable to the risk of a single point of failure due to attack and system break down, etc. Blockchain provides an alternative way of handling access control. If the access control is stored on the blockchain, it is immutable. This makes it hard to be attacked and is available on every node.

Third, the distributed ledger technology grants people the power to control their own data in a distributed way. In a psychology test, the test result data collected usually is highly sensitive and is about someone self. People who take the test ought to have special interests in controlling their own test results data, at least know who has viewed their data. Traditionally psychology test data are managed in a centralized way. In that way, people have very weak or no control of their own test results data. In theory, it may be possible to allow everyone to define their own access control on their own data through the centralized system. However, the central node has the superpower to override the access control data that the user defined. In Blockchain, there is no central point, the controlling of the data is handled in a decentralized way. The access control can be made available on every node. In this way, it is possible for everyone to define their own access control data and have real control over their own data.

Until now, blockchain technology has never been introduced to psychology test data management. The benefits the blockchain technology brings can potentially bring benefits to psychology test data management. It may change the fundamental structure of current psychology test data management by completely remove the centralized authority or by reducing part of the functions of the centralized authority. The problems that the current system faces can all be solved by the change of data organization. This thesis intends to explore this possibility. In addition, it is worth to note that although the solution here focuses on psychology test data management, it also may be of interest for other areas that involve sensitive questionnaire-based data.

CHAPTER 4 ARCHITECTURE

In answering the second research question, which asked for an academic solution, this chapter aims to propose a theoretical architecture that utilizes the benefits of blockchain in psychology test data management and implement it. The intended basic model is shown in figure 4.1. The critical data are stored on distributed blockchain with each node having one copy. Every node verifies each change of the ledger and there is no need to have a central node.

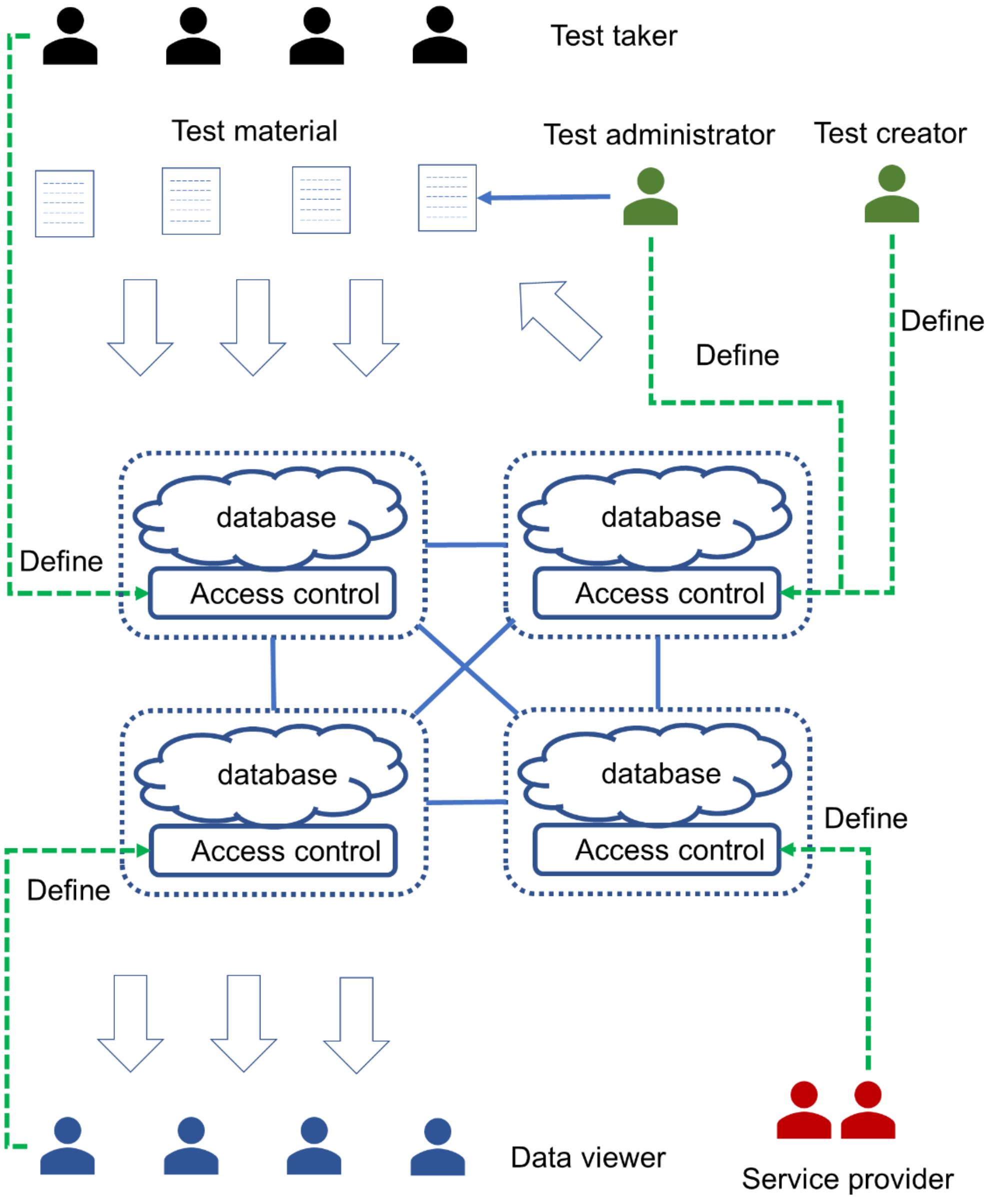


Figure 4. 1. The basic model of psychology test data management using blockchain

In this way, a single source of truth for psychology test data management is possible. Because all the information is chained together, the records stored on the blockchain are immutable. The user can connect to one of the nodes to access the information stored on the blockchain, making it highly available.

The access control of data can be defined by many parties, depending on different scenarios. For instance, the test creator develops the test and need to have control of who has access to the test materials. The test administrator who administrates the testing need to have control of which test taker has access to the test materials. After taking the test, the test taker needs to be able to define who has access to the test results. The data viewer should be able to define access to data if the data need to be transferred to another individual. The service provider should also have control over the data since it provides the service.

An example scenario in a research setting is as follows. A test publisher has developed and published a psychology test. When a researcher buys this test, the publisher grants test administrator access of the test to the researcher. The researcher then can grant access to the test material to the specific test takers. Afterward, data are collected, scored, and stored in the database. By taking the test, the test taker agrees that the results will be available to the researcher. The researcher may share the data with colleagues through the data access control. The University as the service provider also has access to the data. The test taker can verify who has access to the test results. All the transactions will be recorded on the blockchain.

Another example scenario can be in a mental health diagnoses setting. A test publisher has developed and published a psychology test for mental health diagnoses. When a psychologist buys this test, the publisher grants test administrator access of the test to the psychologist. The psychologist then can grant access to the test material to the specific test takers (potential patients). Afterward, data are collected, scored, and stored in the database. By taking the test, the test taker agrees that the results will be available to the psychologist. If the test taker decides to see a different psychologist, the test taker can remove the access of the previous psychologist and grant score access to the new psychologist.

Note that in the above two scenarios, there are different requirements in who can control the test result data. In a research setting, more people can define the access control of the data. In most cases, the data collected is anonymous so there would be fewer problems if more people have control over the data. However, in a mental health diagnoses setting, data are related to each specific individual. The patient needs to ultimately decide who can view the data since it is required by the Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017). Here the solution is focusing on the second scenario, but it can be modified easily to be used in the first scenario.

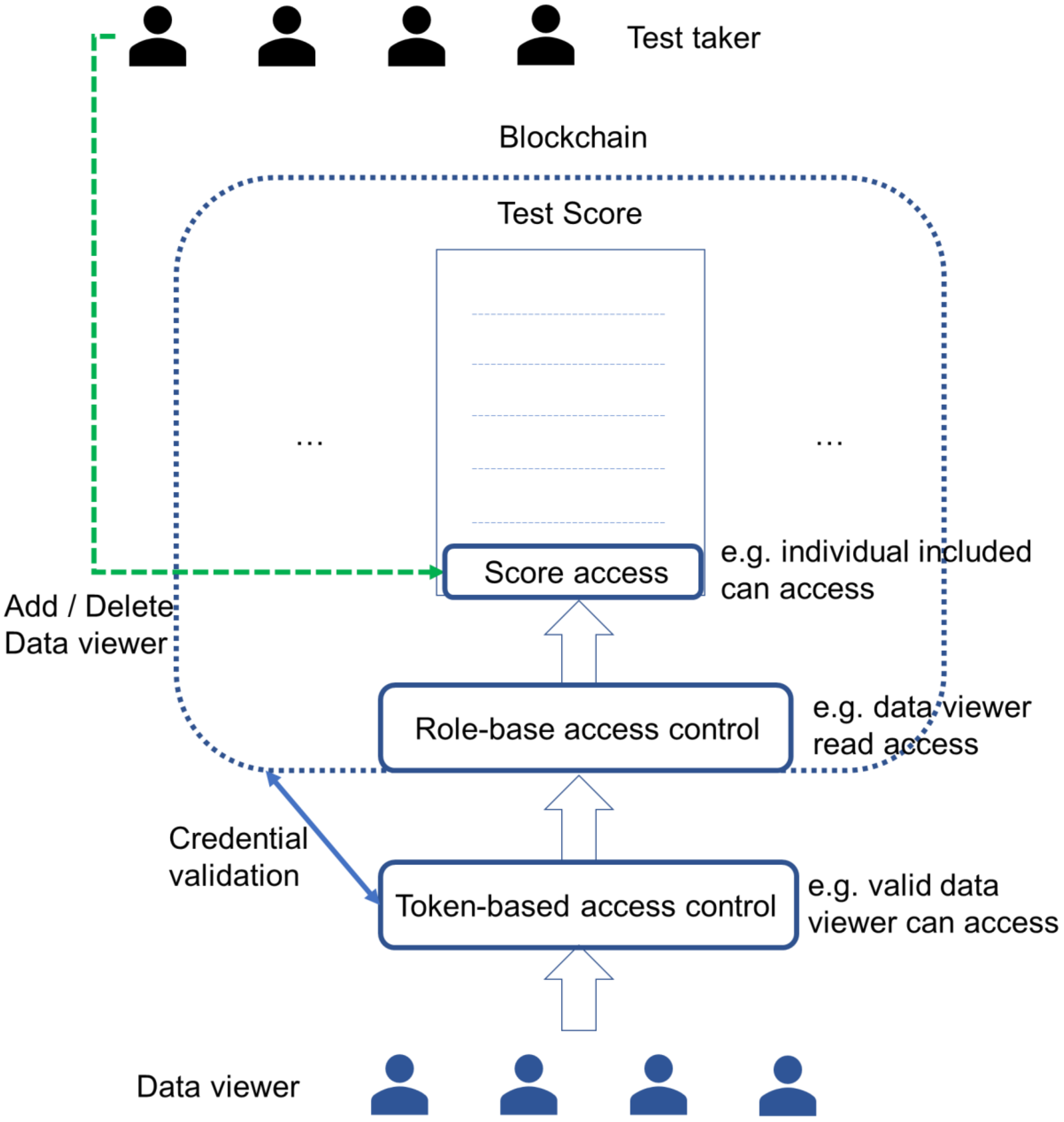


Figure 4. 2. An example of the access control mechanism

To control the data access, a mechanism is adopted. An example of the mechanism is shown in figure 4.2. The test taker can control his/her test score access by adding or deleting a data viewer ID in the score access field of the corresponding test score object. To view the test score, there are three sets of access control rules need to be passed. First, there is token-based access control. It controls a valid user under a type can interact with the blockchain. By providing the correct credential, the data viewer will get a token to access the front-end data viewer module and the web service data viewer module to interact with blockchain. Second, there is role-based access control at the blockchain level. It controls which role has what access to which types of resources. For instance, the data viewer has read access to the test score objects. Third, there is individual-based access control at the individual asset (test score) level. It controls which user has access to the specific asset. For instance, if a data viewer ID is included in the score access field of the asset, then the data viewer can read the test score. Otherwise, the data viewer can not access the test score. In this way, the data are well protected. The same access control mechanism also applies to test material data as well.

For the data storage, generally, there are two ways to store psychology test data management. One way is to store all the data on the blockchain, the other way is to store part of the data on the blockchain and store the rest of the data off the blockchain. Both ways have been used by previous studies in medical record data management, as we have mentioned in the literature review.

The current solution intends to store part of the data on the blockchain and the rest of the data off the blockchain. There are several reasons for that. Generally, there are psychology test materials, test results, and access controls of both the test material and test results that need to be stored. For the test materials, in many cases, they are relatively long (multiple page questionnaires) and require a lot of storage space. The test results include two parts; one is the final score and the other is the raw results. In many cases of psychology tests, the final score includes one or several numbers (e.g., a total score and several subsection scores), which does not require a lot of storage space. However, the size of the raw test results might be large and require a lot of storage space. For the access controls, their size varies but they are relatively small. Because all the data need to be replicated in every node of the blockchain, the more data stored on the block, the slower the overall performance will be. Also, to utilize the immutability and the availability features of the blockchain, important data needs to be stored on the blockchain. Thus, as a balanced solution, the current solution intends to store the final test score, test metadata, and access controls on the blockchain, whereas store the test materials and raw test results off the blockchain and into a centralized database. To connect the centralized database and the blockchain, the hash values of the materials that are stored in the centralized database are also stored on the blockchain.

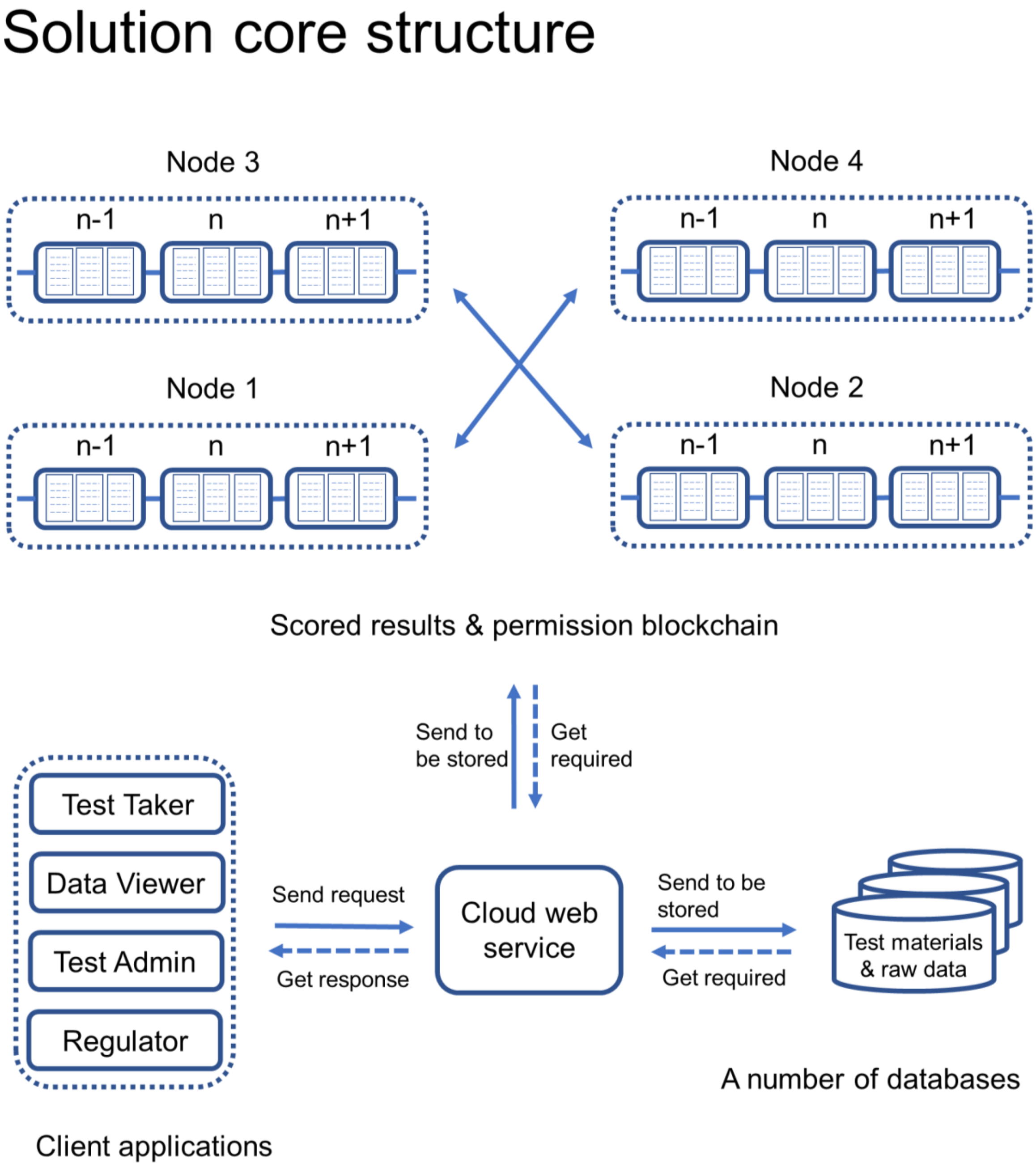


Figure 4. 3. The basic structure of the blockchain solution for psychology test data

The core structure of the solution is shown in Figure 4.3. The final test score and permissions to access to the data are stored on the blockchain which runs on multiple nodes host by different institutions (e.g., clinics), whereas the test materials and raw results are saved separately in centralized databases. There are multiple cloud-based web services that connect the user, the blockchain, and centralized databases. Here to simplify things, one web service and one centralized database are shown. The user only needs to interact with the web service. There are four basic types of the users: test taker who takes the test; test administrator who administrates the test (including test creator); data viewer who views the test score; and regulator who supervises the whole network activity. Depending on the user's role and requests, the web service will execute different functions to provide the required results.

4.1 The architecture of the blockchain

The design of the blockchain in current solution is based on one popular blockchain infrastructure, namely the Hyperledger Fabric (The Linux Foundation, 2017), although the design here may be able to be applied to other blockchain infrastructures with some modifications. The designed blockchain includes three basic elements and a set of permission control rules. The three elements are participants, assets, and transactions. The permission control rules define who has access to what resources under what conditions.

Participants are the users of the network. There are four different types of participants. Every user interacts with the blockchain network as one type of participants. Based on their role, the participants may own assets, or are able to submit transactions to change the status of the assets or resources within the network. As mentioned above, the four types of participants are test taker, test administrator, data viewer, and regulator. Participants' accesses to different resources of the blockchain are controlled by the permission control rules.

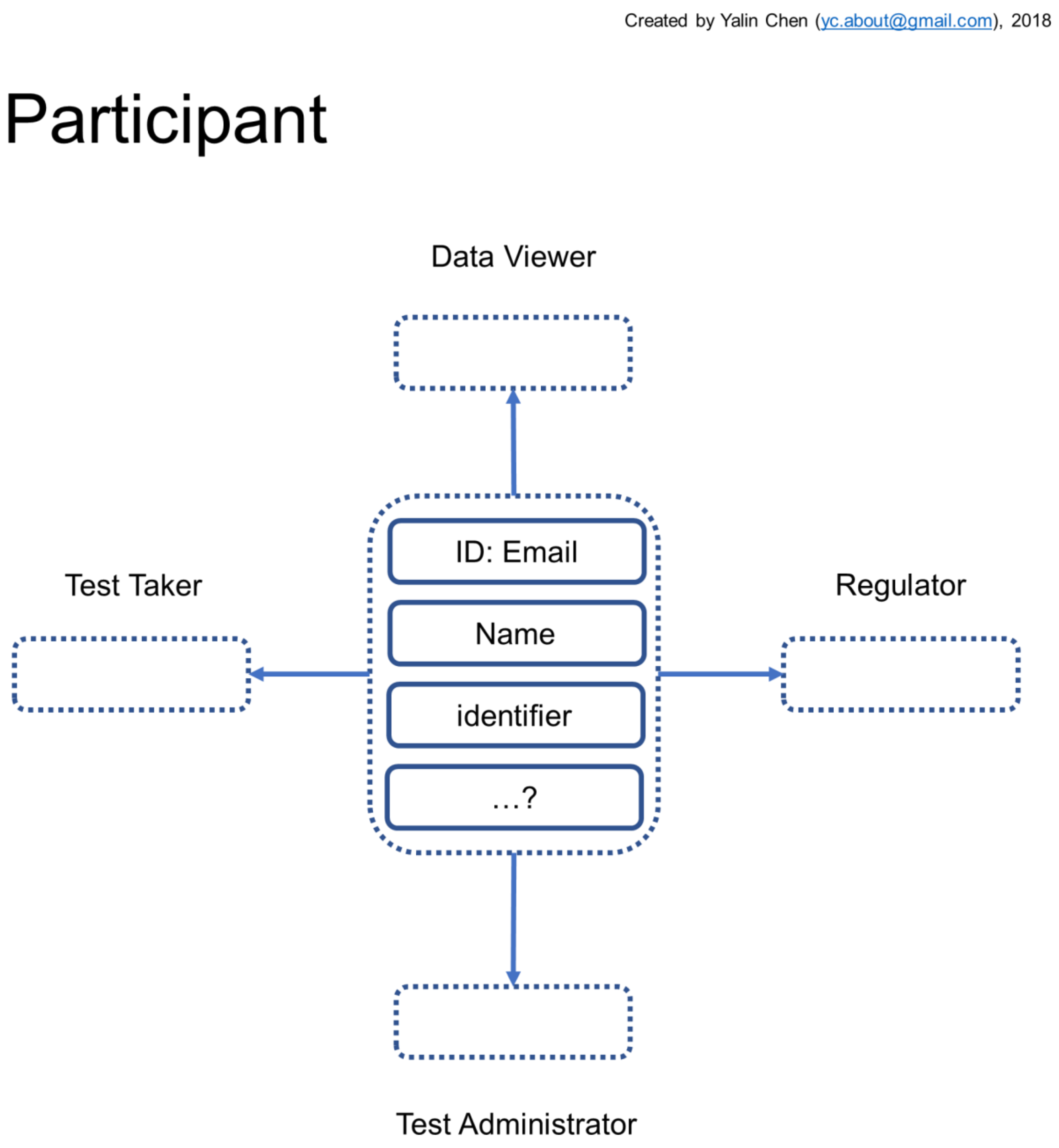
Assets are goods, services, or properties within the network. In the current solution, there are two basic types of assets. They are psychology tests and psychology test scores. A test asset contains the test metadata and the hash value of the test materials that are stored in a centralized database. A test score asset is generated after a test is completed by a test taker and the raw test results are graded by a test administrator. It contains the graded results, test score access, and test access, etc.

Transactions are the mechanisms by which participants interact with assets. For instance, a participant can create an asset, or transfer an asset to another participant. In the current context, transactions either create new assets or change certain fields of target assets. For example, a complete test transaction which is submitted after a test is completed and the raw results are graded will generate a new test score asset. A change test score access transaction submitted by the owner of the test score will change the test score access field of the target test score asset.

The permission control rules are the rules that define which participant can do what operations on what resources under what conditions. For instance, the test takers should only have the read permission on tests that are available to them. They need not have update permission on the test materials, and they need to only have read permission on their own test score assets.

The participant, asset, and transaction elements define the basic structure of a functional blockchain network. The permission control rules provide protection of the contents stored on the blockchain. The current solution designed these three elements and a set of permission control rules in the following manner.

The participant of the current blockchain solution is shown in Figure 4.4. There are four types of participants: test taker who takes psychology tests; test administrator who administrates psychology tests and score the raw results; data viewer who views psychology test data; and regulator who oversees the whole process. All the participant types have several fields: email as their ID, their name, identifier, etc. The identifier field is used to store the hashed password of the current participant. It is used to authenticate the user within the system. With the password as one field of the participant, there is no need for an independent component to hold participant login information. The hash value of the password rather than the password itself is stored to guarantee security for the password storage. Other fields may be added to the participant object depending on the role if needed.


Created by Yalin Chen (yc.about@gmail.com), 2018

Participant

Figure 4. 4. Participants of the blockchain solution

There are two types of assets, tests, and test scores, as shown in Figure 4.5. The test refers to the metadata of a given psychology test, which includes a hash value of the test contents. The creator field of the test defines the creator of the current test. The director field defines which test administrator oversees the current test. This director administrator controls which test administrator has administrator access to the current test asset through the testAdmin field. This was done by referring to the testAdmin field in corresponding permission control rules. The permission control rules grant permission to personals who are included in this field. The test director can transfer the director position to another test administrator by submitting a transaction. There is another special field named test access. This field controls which test taker has the permission to view the metadata of the test. The test administrator of the current test can update this field (i.e., add and delete a certain user) by submitting a transaction. When a test taker is added to this field, the test taker can view the metadata of the test and complete the test.

For the test score asset, it contains information about the test taker of the current test score, the test of the current test score, the final graded score, the subsection scores, and the access to the current test score, etc. The score access field defines which data viewer is able to view the current test score asset. The test taker of the current test score can update this field (i.e., add or delete a certain user) by submitting a transaction. In this way, the test taker has control over who can view which test score of the test taker. To increase flexibility, both the test asset and test score asset have a field named note. This is used to hold extra information about the test or test score. For instance, the test score note can be psychologists' notes and recordings concerning client/patient statements and behavior during an examination.

As can be seen from above, the access control of a specific asset is defined within the asset itself, both for the test asset and for the test score asset. In this way, it is possible that the access control is verified at the time accessing the asset. For the test score asset, who gets to define the access field varies depending on different scenarios. For instance, in a research setting where the data are anonymous, there may be more parties can control the access. However, in a mental health diagnoses setting, fewer parties can control the access field because the data are linked to specific individuals.

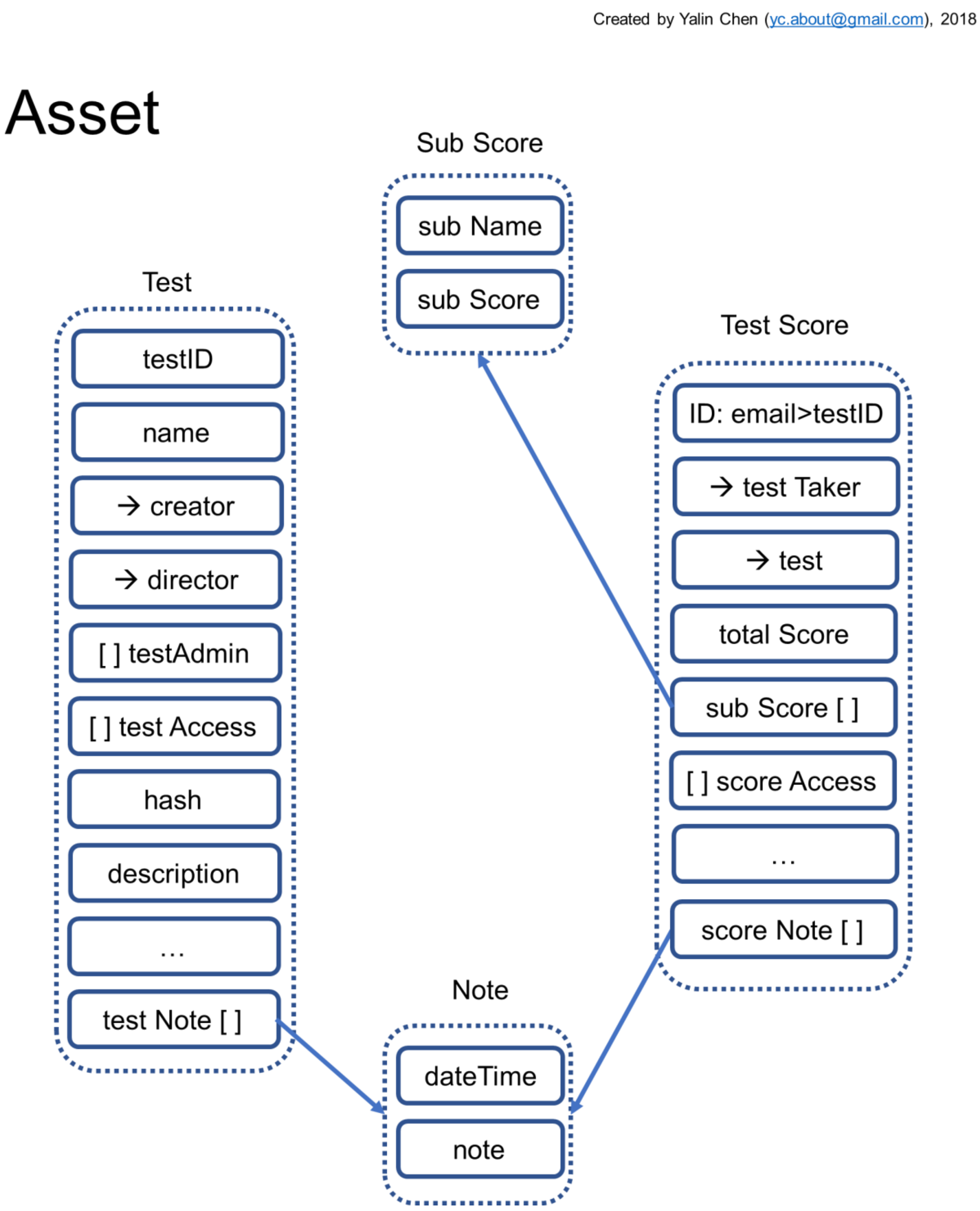


Figure 4. 5. The asset of the blockchain solution

There are seven types of transactions, as shown in Figure 4.6. They are update test director, update admin access, update test access, update test notes, complete test, update score access, and update test score notes. The first four transactions operate on the test asset. The last three transactions operate on the test score asset.

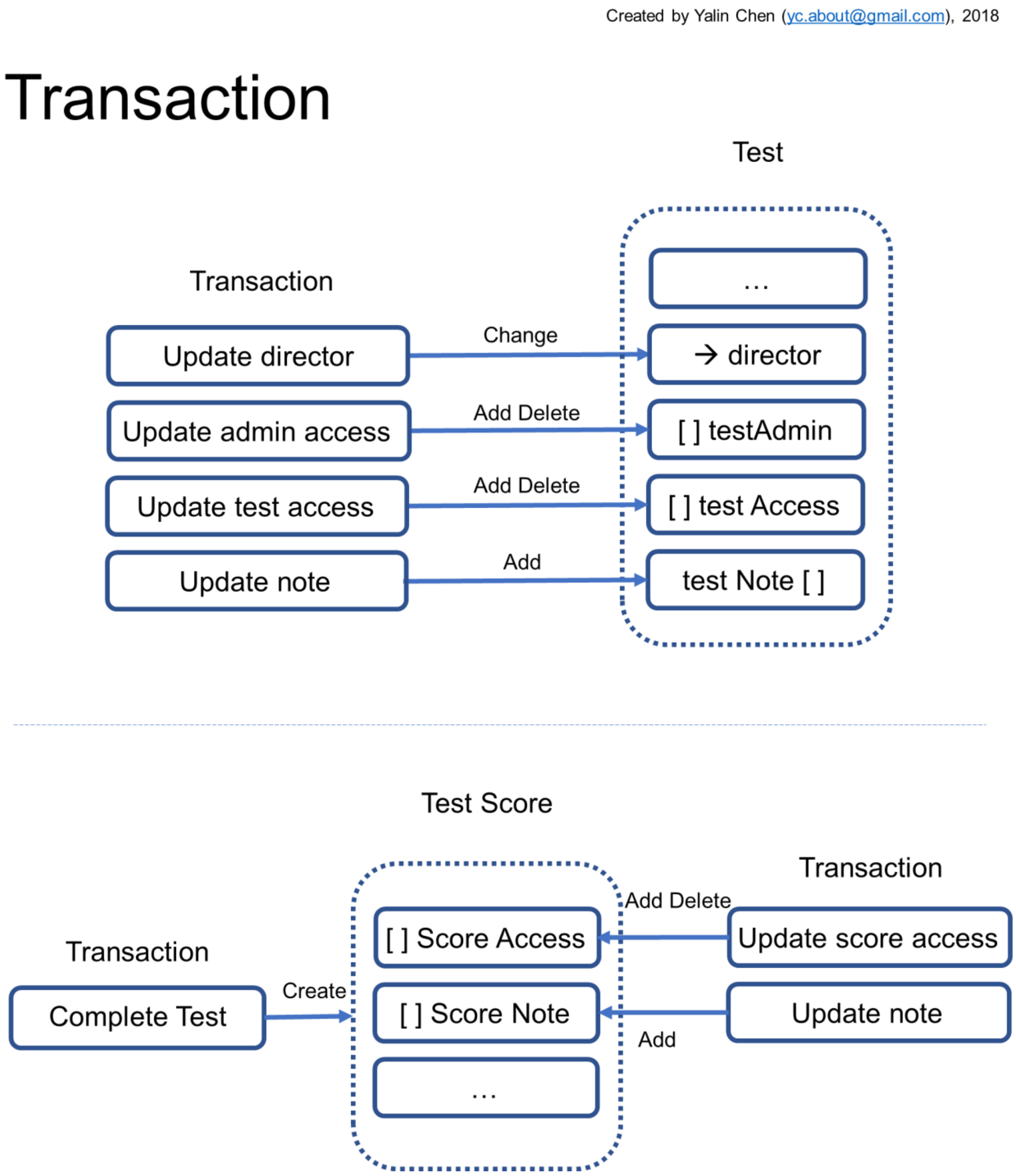


Figure 4. 6. Main transactions for the blockchain solution

The update director transaction changes the director of the test. It can change the value of the director field of the test asset. To avoid conflict, there is only one director for a given test asset at a given time. The test director can submit the update admin access transaction. The update admin access transaction can add or delete test administrators for the target test. It can change the values of the test administrator field of the target test asset. The test administrator that is added will be able to manage the test asset (e.g., grant read permission of the test to the certain test taker, grading the raw test results, etc.), whereas the test administrator who is not added will not be able to manage the test asset. The update test access transaction can add or delete test takers who have access to the test. It can change the values of the test access field of the test asset. The test taker that is added will be able to view the test asset, whereas the test taker that is not added will not able to view the test asset. The update test notes transaction can add test notes to the target test. It can change the values of the test note field of the test asset.

The complete test transaction creates a new test score asset. The update score access transaction can add or delete data viewers that have access to the test score. It can change the values of the score access field of the test score asset. The data viewer that is added will be able to view the test score asset, whereas the data viewer that is not added will not able to view the test score asset. The update test score notes transaction can add notes to the target test score. It can change the values of the test note field of the test score asset.

There are certain permission control rules that apply to the different roles of participants, as shown in Table 4.1. For instance, a test taker has create and read control of his/her own profile, has read permission on the test asset if the test taker is in the test access field of the test asset, has update permission on the own test score asset (represented as own in the table) while submitting an update test score access transaction, has create permission on update test score access transactions, has create permission on add test score note transactions, and has read access to the historical records where the test taker is a participant. Test director can create an update test director transaction or create an update test admin access transaction (represented as director in the table). The data viewer has read permission on the test, has read permission on the test score if is permitted, has create permission on the test score note transaction, and has read permission on own operation history. The regulator only has read permissions on most of the resources. Generally, the permission rules can be classified into two types. One type of permission rules is role-based access control rules. They define access control based on different participant roles. The other type of permission rules is reference-based access rules. They define access control by referring to certain fields of certain objects. The permission control rules control all the resources within the network. The user acts as a certain type of participants and going through the permission control rules to interact with the network. In this way, certain resources within the network are well protected.

Table 4. 1. The permission control table for the blockchain.

Resources	Test Taker	Test Admin	Data Viewer	Regulator
Participant				
Own profile	CR	CR	CR	CR
Other profiles	-	-	-	R
Asset				
Test	R (test access)	CR U (TA)	R	R
Test Score	R (own) U (TSA)	C (CT)	R (score access)	R
Transaction (Tx)				
Test Director	-	(director) C	-	R
Test Admin Access	-	(director) C	-	R
Test Access (TA)	-	C (admin access)	-	R
Test Note	-	C (admin access)	-	R
Complete Test (CT)	-	C	-	R
Test Score Access (TSA)	C	-	-	R
Test Score Note	C	-	C	R
History	R(own)	R(own)	R(own)	R

Note. C = Create, R = Read, U = Update

4.2 The architecture of the web service

The web service component mainly connects the user from the front end, the blockchain, and the non-blockchain databases. It receives user requests from certain APIs. Depending on the requests, it communicates with the blockchain, non-blockchain databases, or both to provide proper responses to the user requests.

The overall architecture of the web service is shown in Figure 4.7. The current solution intends to build multiple cloud-based servers running through popular cloud service such as Google Cloud or Amazon Web Service (AWS) to guarantee accessibility. For simplicity, here one example of cloud web service is presented.

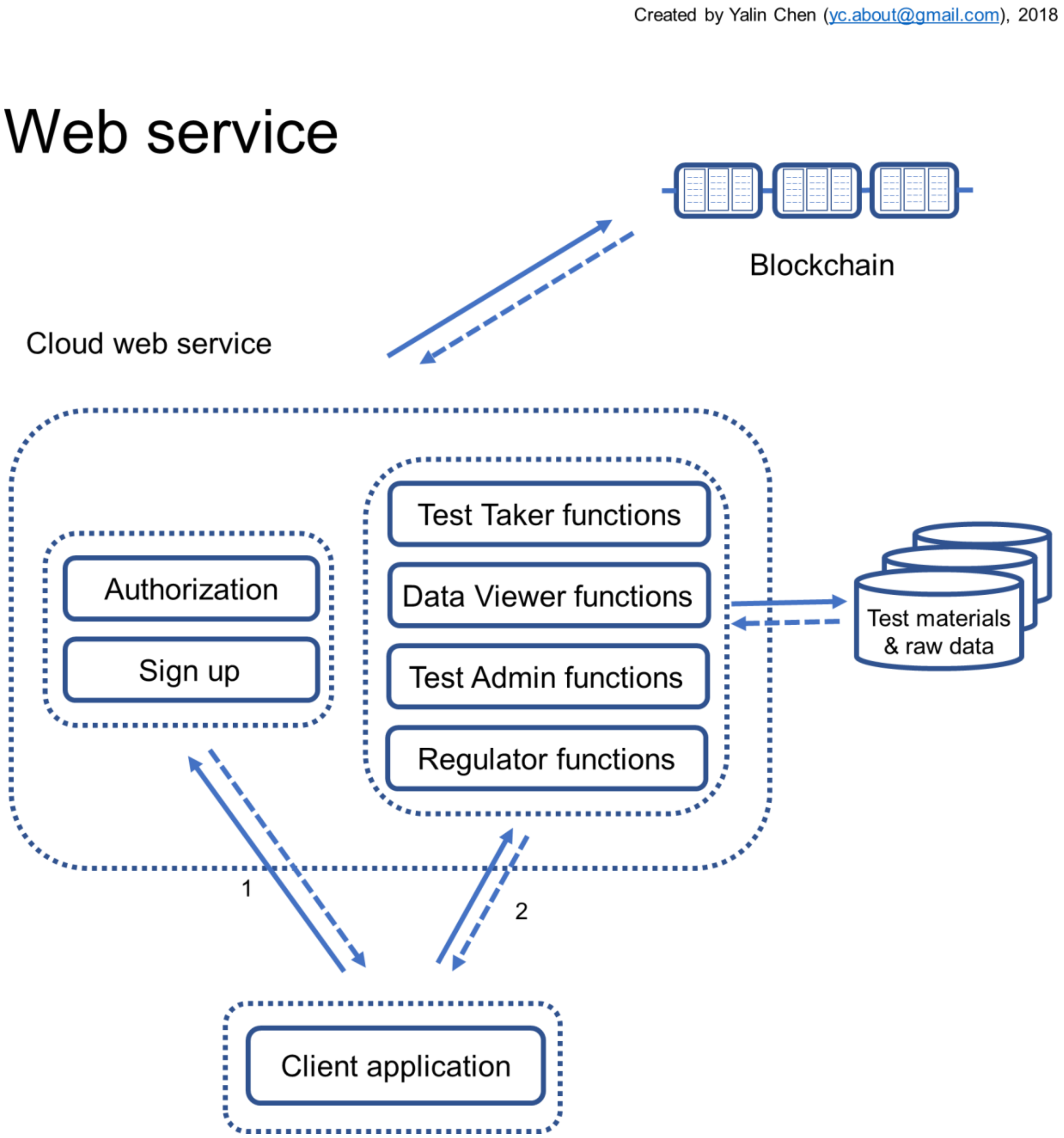


Figure 4. 7. The web service component of the solution

As can be seen from Figure 4.7, first, the user needs to be authenticated to the server. This includes user sign up if the user does not exist in the network and user authorization if the user exists. After the user authorization, a specific blockchain participant role (i.e., test taker, test administrator, data viewer, or regulator) is bind to the user. Depending on the role, the web service will communicate with the blockchain or centralized databases through the functions that corresponding to the role. Thus, the web service contains three basic modules: a module that handles user registration and authorization (i.e., login), a module that handles communication with the centralized database, and a module that handles communication with the blockchain. In the following, these three modules will be explained in detail.

The user registration and login module receive the user sign up and login requests and executes the corresponding functions. Figure 4.8 shows the user sign up workflow. For the user sign up, first, the client collects the user sign up information and then send them to the web service. The web service receives the registration information and then hashes the password. Afterward, the web service calls the create participant API of the blockchain and sends the required participant information including the hashed password to the blockchain. The blockchain creates the participant object with the encrypted password as one field of the corresponding participant. Subsequently, the web service calls the issue identity API of the blockchain to issue an identity to the participant created. Then, the web service creates a business network card using the identity key, the connection profile and the metadata. Finally, the web service imports the business card created to the blockchain for further use. All the further requests from this participant will be made using this business card. The business card is a critical part of the access control of the blockchain resources. The access control rule defined for the network operates based on the identity that is acquired from the business card.

Figure 4.9 shows the workflow for the user login process for the web service. First, the client collects user login information and send them to the web service. Then the web service receives the information and hashes the password. Afterward, the web service calls the get participant API of the blockchain using the username and role provided to acquire the corresponding participant information. Then, the web service compares the hashed password the current user provided with the hashed password acquired from the blockchain. If they match, a JSON web token (JWT) will be created including the username and the role as payload. The JWT will expire after a certain period. For every subsequent API call from the front end, this token needs to be used as part of the call. For all the subsequent calls received by the web service, this JWT token will be verified first. If the token is valid, then the web service will use the username decoded from the JWT to find the corresponding business card and interact with the blockchain using the card, or the web service will interact with the centralized database if needed.

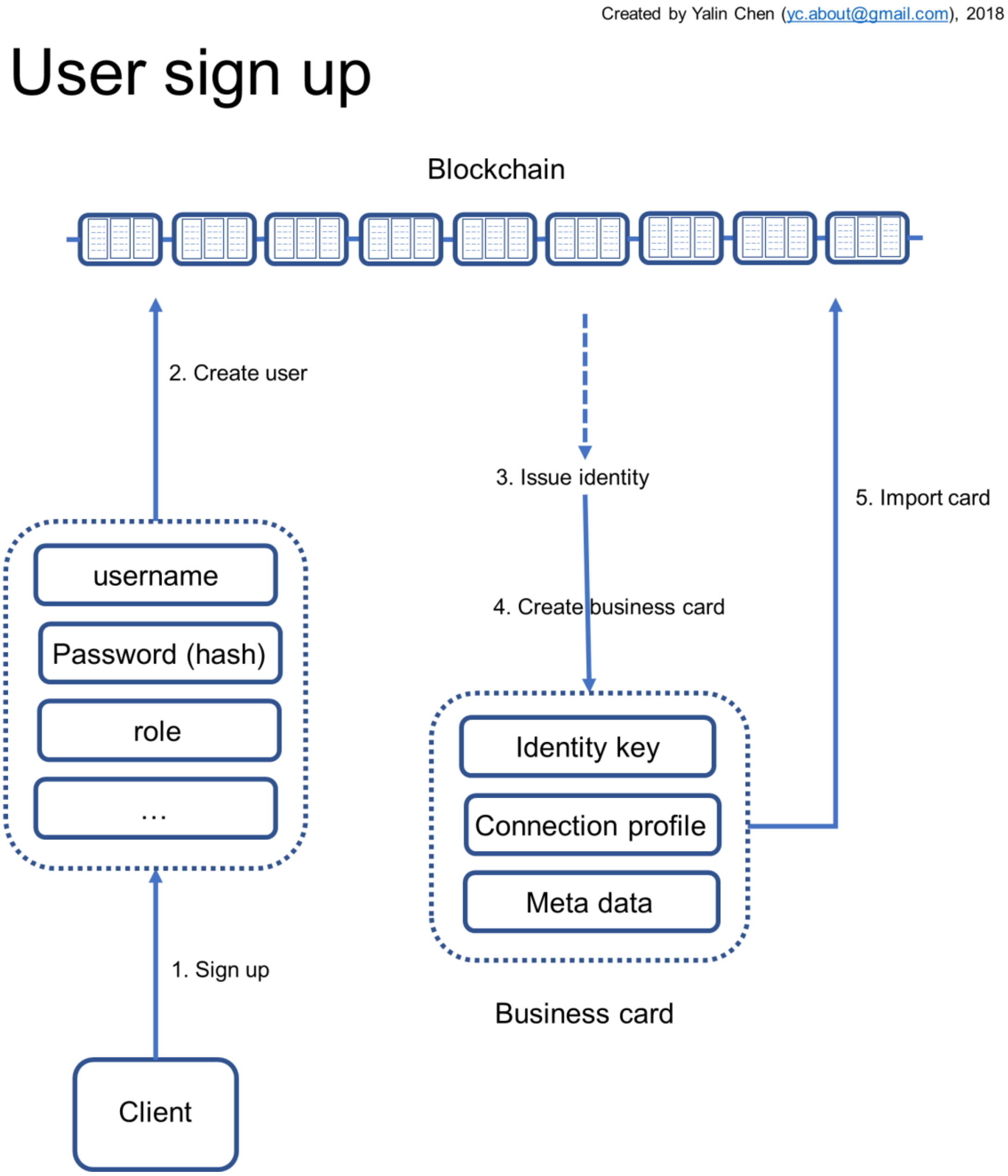


Figure 4. 8. The user sign up workflow

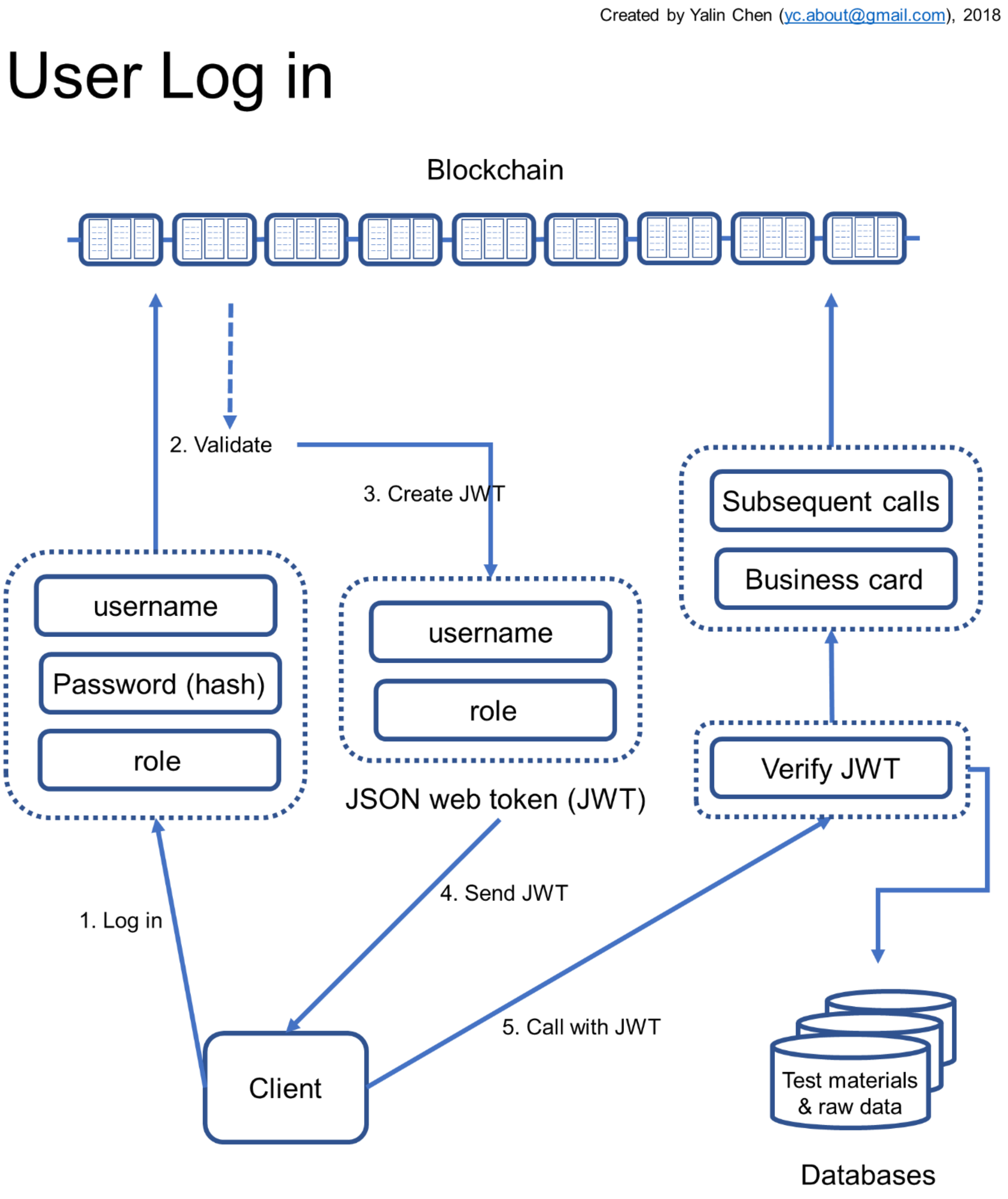


Figure 4. 9. The user log in workflow

The module handles communication with the no-SQL database mainly answers API calls from the front end and execute certain operations on test materials and raw test results. Different requests send to different API paths. Based on different API paths, the web service calls different functions and send the returned values of corresponding functions as the responses to the requests. No-SQL database was chosen because the major purpose for the database is to store the test materials and raw test results. In traditional SQL databases, the structures of the to be stored materials need to be pre-defined as tables before they are stored. Psychology test materials have all kinds of different structures. Thus, it is very hard and has less flexibility to predefine a structure table to hold all kinds of different test materials and raw data. It is more convenient to keep the data structure for test materials and raw results in JSON format. In this sense, a no-SQL database serves the current purpose better than a traditional SQL database.

Table 4. 2. Web service functions for the no-SQL database based on different roles

Role	No-SQL database functions
Test taker	Read test materials Create test results
Test administrator	Create test materials Read test results Update test results
Data viewer	Read test materials Read test results
Regulator	-

Based on different participant roles, different functions are provided. Table 4.2 demonstrated the proposed web service functions for different types of participants. Specifically, a test administrator can call the corresponding API path to add the test materials of a test to the database. A test taker can retrieve the test materials from the database to complete the test if the test taker has access to the test (i.e., controlled by the blockchain). After the test is completed by a test taker, the raw results will be stored in the database. It will be flagged as not scored. Then, the test administrator of the test can retrieve unscored raw test results, score them, update the score status, and submit the final score to the blockchain. The test taker can then check his score and update who has access to the test score asset. When viewing the test score asset, the data viewer also can view the test materials and raw test results if they are required.

The no-SQL database does not work on its own. Otherwise, it inherits the drawbacks of the centralized system has. The no-SQL database is connected to the blockchain in certain ways to guarantee that data stored in the no-SQL database cannot be modified freely. For instance, after a test administrator adds the test materials of a test to the no-SQL database, the added content of the test materials is hashed, and the hashed value is passed to the blockchain to store as a field of the test asset. In this way, any malicious attempt to modify the test materials can be easily detected. Also, in this sense, the regulator does not have to view the test materials, given that the hash value stored on the blockchain can be used to identify the correct test materials if it is needed.

The module that handles communication with the blockchain mainly answers API calls from the front end to execute certain operations on the blockchain. Similar to the module handles communication with the no-SQL database, different APIs are provided to answer different requests. Based on the APIs, the web service will execute different functions and send the returned values as the responses to the requests. Different functionalities were provided based on the role of the participant. Table 4.3 demonstrated the proposed web service functions for different types of participants, in addition to the user registration and login functions mentioned above. Specifically, the test taker can create and read the test takers own profile from the blockchain, read test metadata if has access to, read own test score, update own test score access, and create test score note through the functions. The test administrator participant can create and read own profile, create a new test, grading test results, control which test taker has access to the test, and add a note to the test through the functions. If the test administrator is the director of the test, he/she can control which test administrator can administrate the test. The current director of the test can also transfer the directorship to another test administrator. The create test function also bridges the blockchain with the no-SQL database. As also mentioned above, in creating the test, after the test administrator creates and stores the test materials in the no-SQL database, the hash value will be passed. The create test function takes the passed hash value, combine it with other metadata from the test administrator and creates a test object on the blockchain. The data viewer can create and read own profile, read test, read test score if has access, and create test score notes through the functions. The regulator has the functions to create and read own profile and read transaction histories.

Table 4. 3. Web service functions for blockchain based on different roles

Role	Blockchain functions
Test taker	Create & Read own profile Read test Read own test score Update test score access Create test score note
Test administrator	Create & Read own profile Create & Read test Create test score (after grading) Update test director Update test administrator access Update test access Create test note
Data viewer	Create & Read own profile Read test Read test score (access) Create test score note
Regulator	Create & Read own profile Read transaction history

4.3 The architecture of the front end

The front end defines the interface for the user to communicate with the blockchain and the centralized database via the web service. The front end collects the user inputs and sends them to the web service via different API paths that were designed. The front end also receives the responses from different APIs of the web service and display them properly to the end user. Depending on the role that the user acting as, different interfaces are provided.

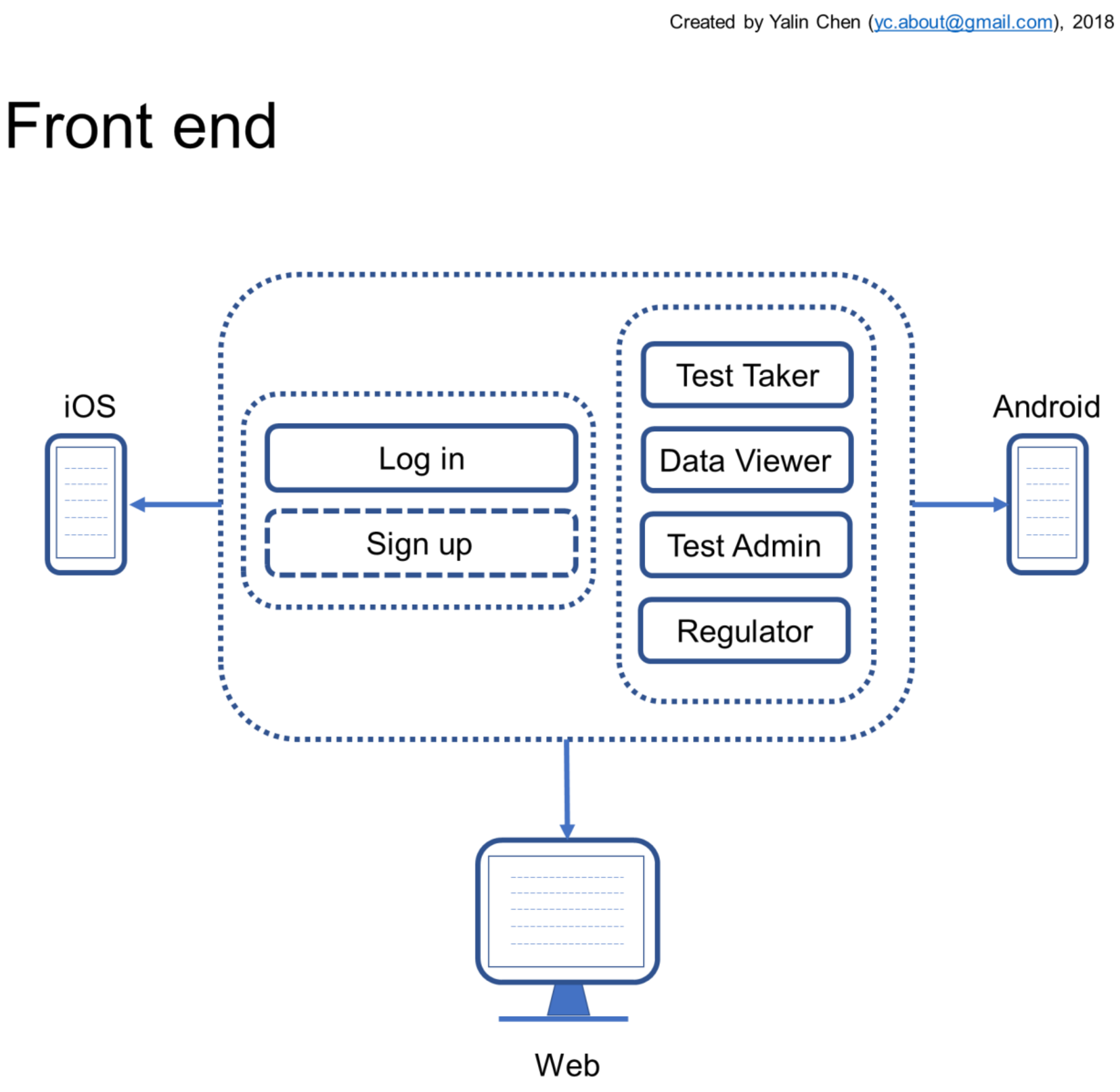


Figure 4. 10. The front end of the solution

The architecture of the front end is shown in Figure 4.10. The front end involves a login in component to authorize the user, a sign-up component if the user does not yet exist in the network as a participant, and a third component which contains available functions depends on the role of the user. The front end should be available on all the three platforms, namely Web, iOS, and Android. Here the focus is on the web platform because the computer screen is more suitable to display a multi-page questionnaire. However, when the mobile ends are needed, tools (e.g., Ionic or React Native) are available to easily transfer the web end to the mobile ends.

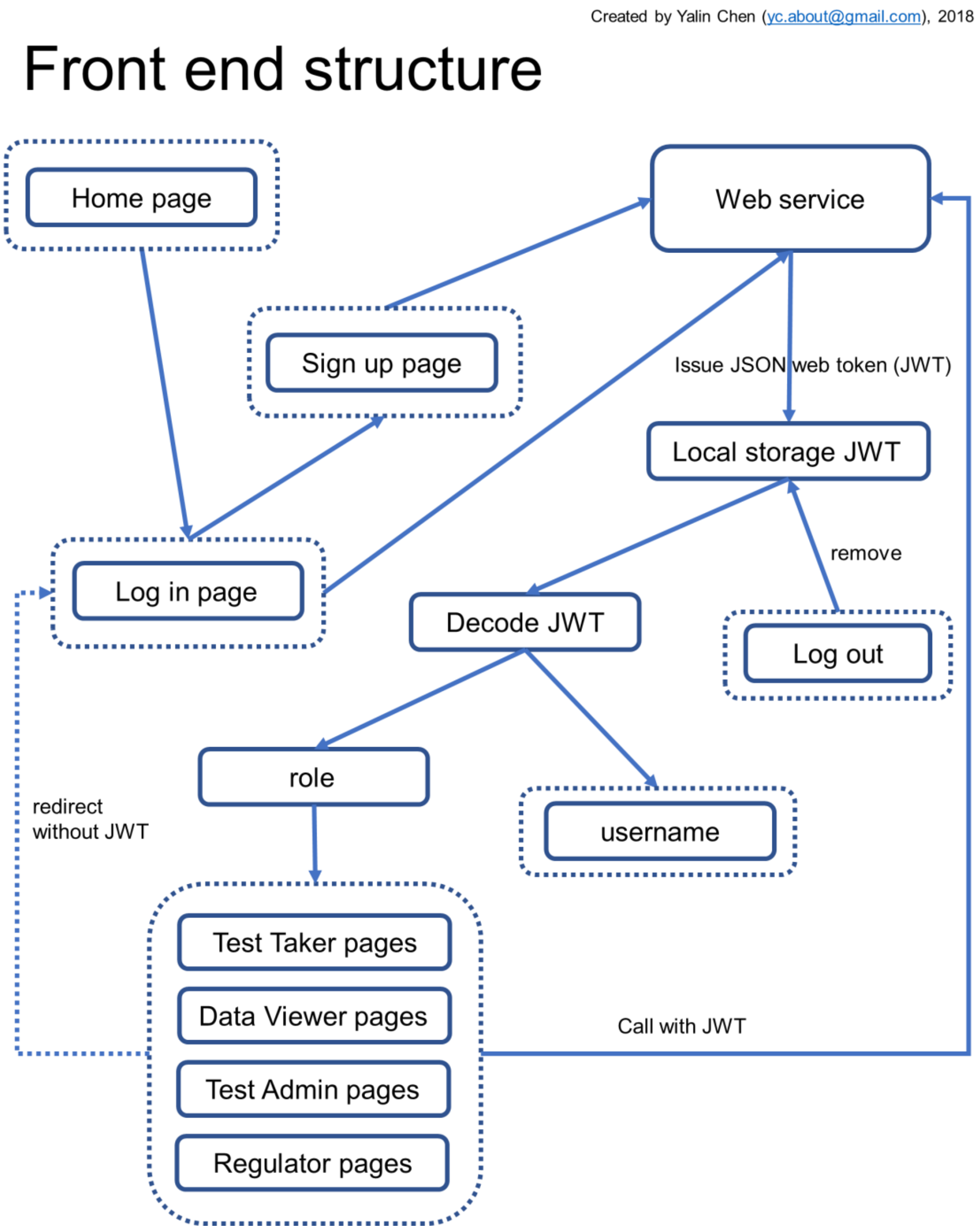


Figure 4. 11. The front end structures

Figure 4.11 shows the organization of the front-end. Without login, users can check the features of the network on the home page, enter the user registration page, and enter the user login page. To access more functionalities, users need to log in. As described in the web service part, when the login information are sent to the web service and they are successfully verified by the web service, a JWT token is issued to the front end. The front end receives the JWT token and stores it in the local storage. For every subsequent API call from the front end, the JWT token is sent as part of the call. After getting the JWT token, the front end decodes the JWT token to get the participant role and username information. Based on the role information contained in the JWT token, the front end renders different functionalities to the current user. For instance, the test administrator has access to create a new test module and score raw test results module, and a test taker has access to the complete test module, etc. There is also a user log out function. When activated, it deletes the JWT token from the local storage. Without the JWT token, any attempt to enter the protected pages will be redirected to the login page for user authentication.

To handle the creation, display, and completion of tests and view raw test results, a proper library is needed. There are a few requirements for this goal. First, the created test materials and the generated raw test results need to be sent to the web service for storage in JSON format. Thus, a straight forward way would be that the front end takes user inputs and convert them into the JSON format and then send them to the web service, both for questionnaire creation and questionnaire completion. Of course, the end user needs to see the questionnaire in a traditional questionnaire display rather than a JSON format object. Thus, for questionnaire creation, the front end needs to provide an interface for the test administrator to create a questionnaire in a beginner-friendly way and convert it into JSON format afterward. For questionnaire completion, the front end needs to take the JSON object and display it as a user-friendly questionnaire. Also, the front end needs to allow user inputs while the questionnaire is displayed, and the user inputs need to bind with corresponding questions. Second, the viewing of the raw test results requires to retrieve and display the questionnaire and the raw results from the web service, which are stored in the JSON format. Thus, the front end needs to get the JSON formatted questionnaire and raw results and convert them into a user-friendly display. Also, the display needs to be read-only. Because this part is not the main focus of the current solution, a third-party library was used.

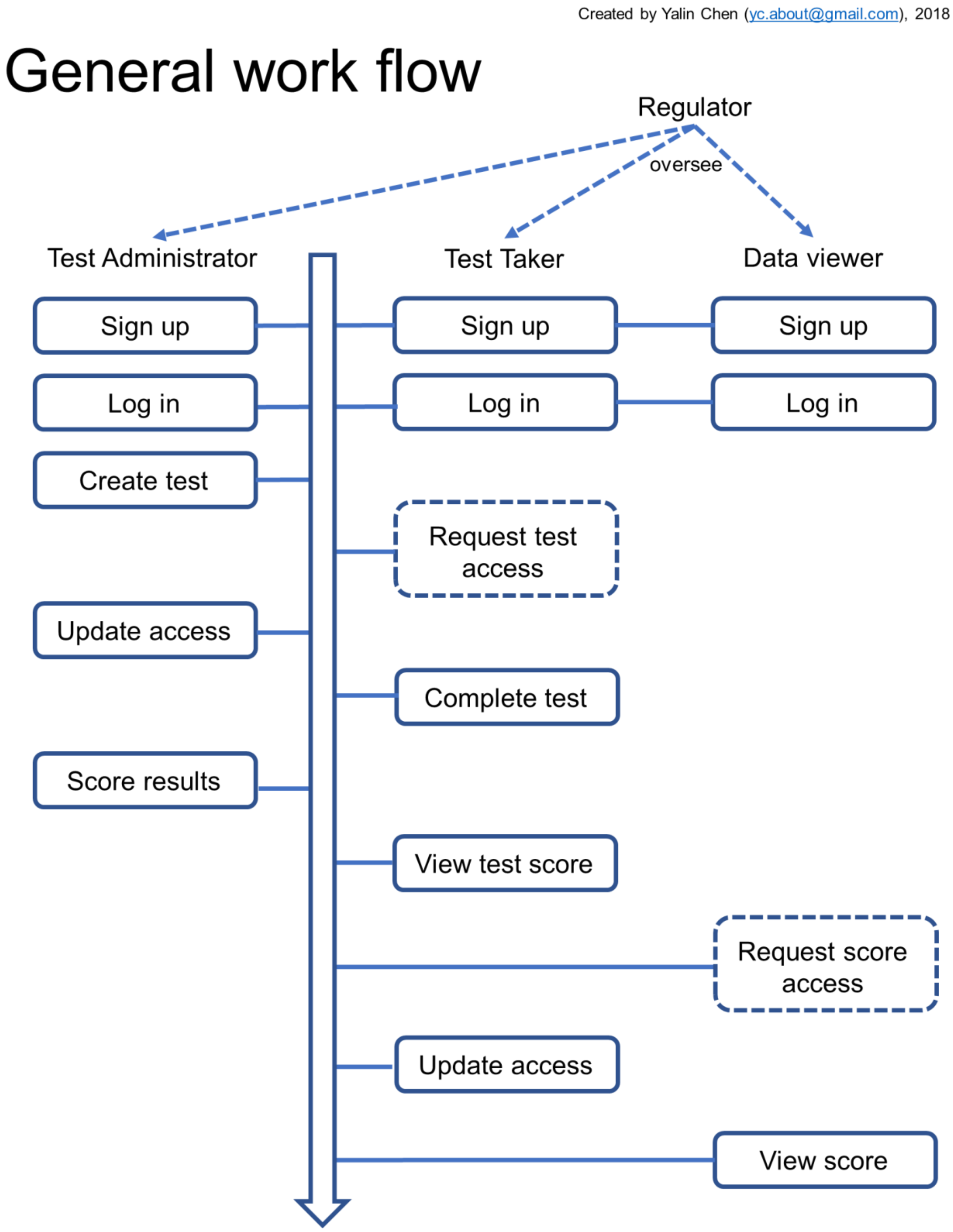


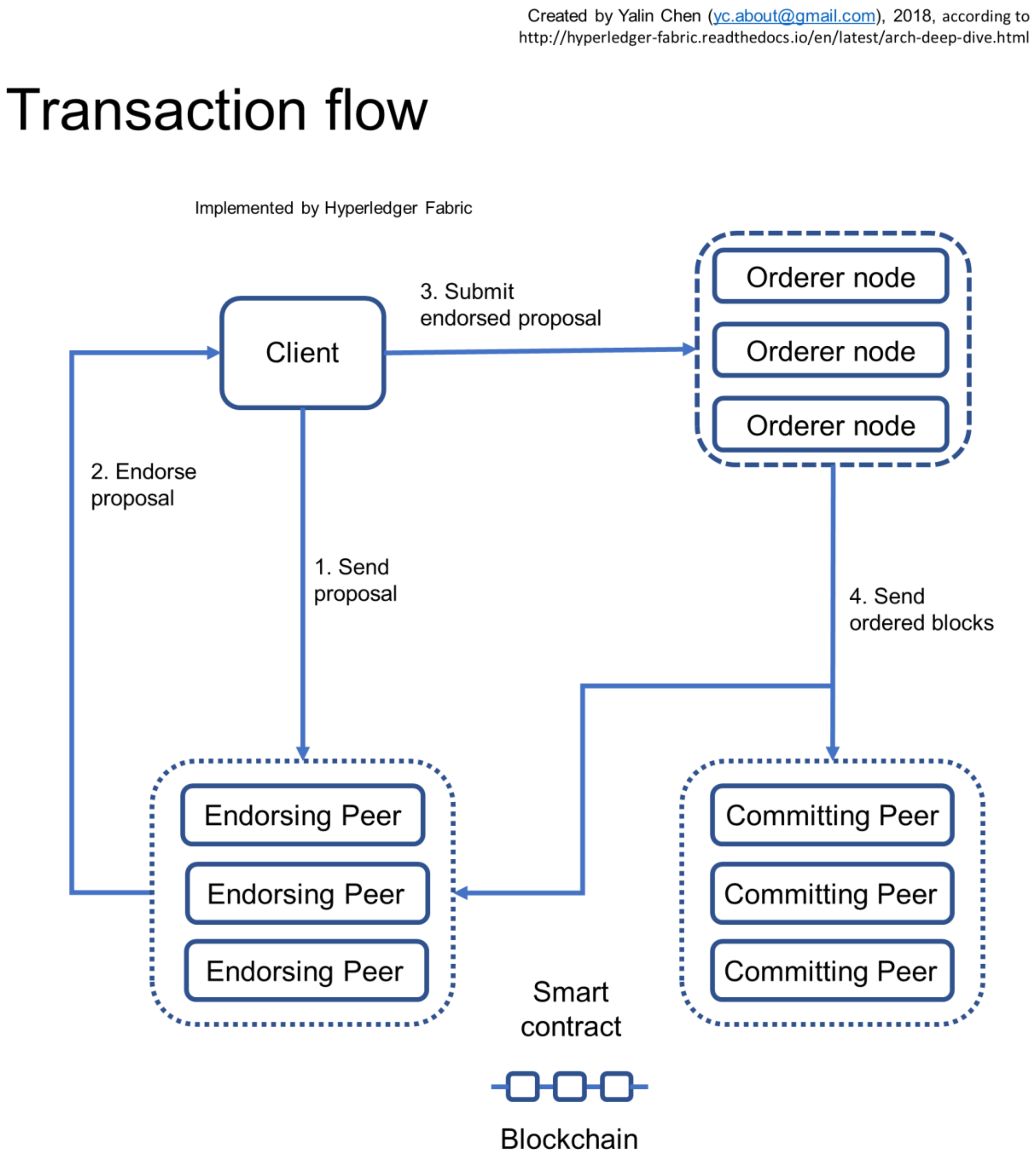
Figure 4. 12. The general workflow of the solution

Overall, a sample workflow is shown in Figure 4.12. First, the test administrator needs to register and log in. Then the test administrator has access to the create new test module. The test administrator can build a test using the embedded questionnaire builder. After the test is built, the test content is saved to the centralized database first and then the meta including the hash value of test content and the unique ID key in the centralized database are saved on the blockchain. On the test taker side, first, a test taker needs to register and sign in to use the system. If a test has already been created by the test administrator in the system and the evaluator of the test taker determines that the test taker needs to take that test, a request can be made to the test administrator. The test administrator then grants test access to the test taker through the corresponding transaction module. At this time, the test taker can see and complete the test within the test taker's own account. After the test is completed, the raw test results are saved to the centralized database. Afterward, the test administrator can retrieve the results, score the results and submit the final score to the blockchain. After score submission, the test taker can see the test score in the test taker's own account and can take control of which data viewer can view the test score. If a data viewer wants to view the data, a request needs to be made and the test taker can add the ID of the data viewer to the test score access field through the update test score access transaction. Afterward, the data viewer can view the test score. The test taker and data viewer can add notes to the test score if there are special conditions that need to be considered after the test score asset has been created. The test director can transfer directorship of a test to another test administrator and can add or remove test administrators for the test after the test has been created. The test administrator of the test can add test note if there are special conditions after the test is created. The regulator oversees the whole process.

4.4 The implementation

The implementation of the solution involves a group of different techniques. The whole solution was developed in the Linux-Ubuntu (16.04 LTS) environment. The implementation of the blockchain used the Hyperledger Fabric infrastructure of the blockchain via the Hyperledger Composer (The Linux Foundation, 2017). The implementation of the web service used the Nodejs. The implementation of a centralized database used the MongoDB database. The implementation of the front end used the Reactjs. In the following, we explained in more details.

The Hyperledger Fabric is an open source project hosted by The Linux Foundation. It has a modular architecture, where different components (e.g., peer nodes, certificate authority, and orderer nodes, etc.) are relatively independent of each other. It allows smart contracts named “chaincode” to run and the chaincode defines the logic of the system.



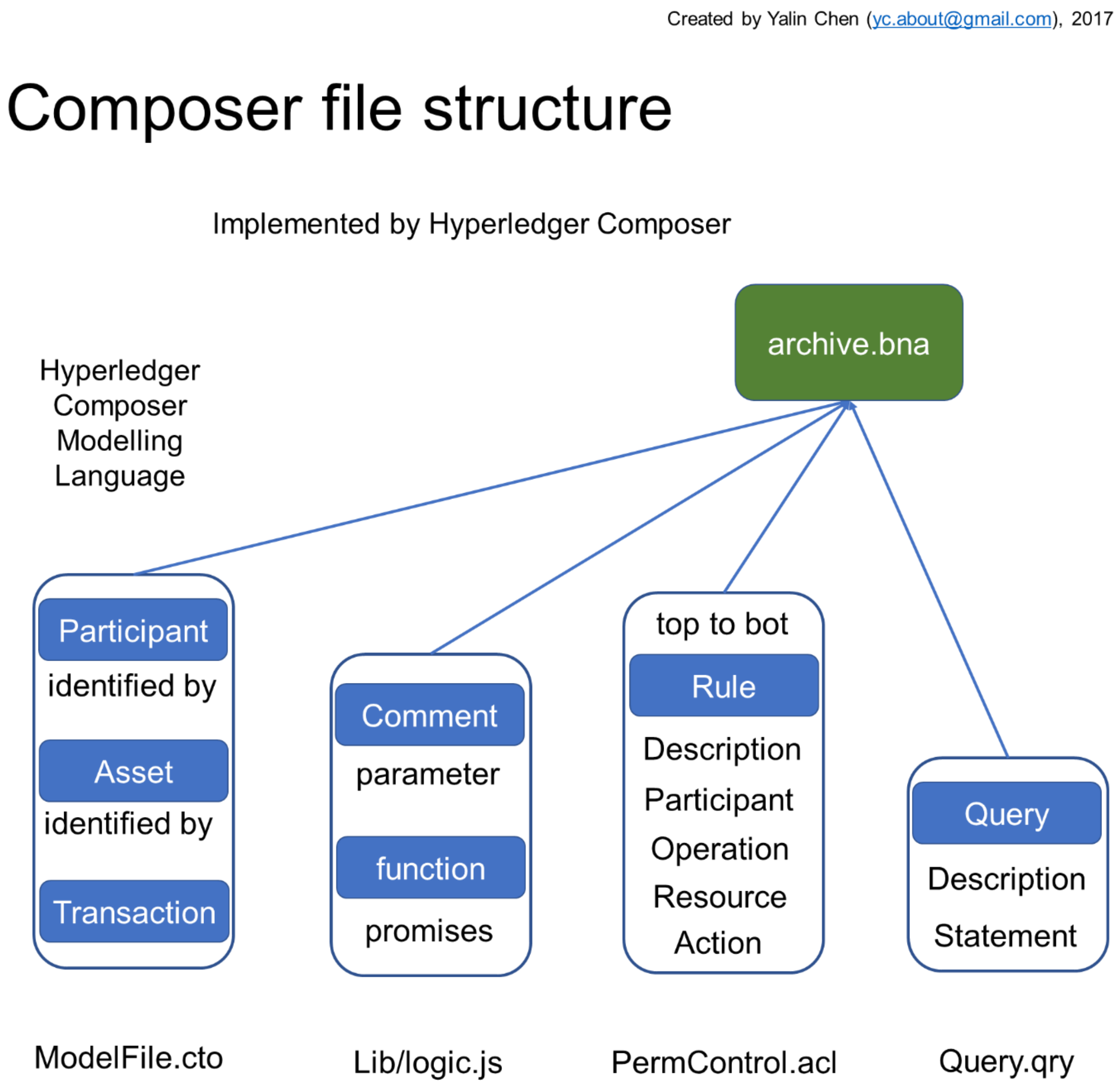
Created by Yalin Chen (yc.about@gmail.com), 2018, according to
http://hyperledger-fabric.readthedocs.io/en/latest/arch-deep-dive.html

Figure 4. 13. The basic transaction flow adopted by Hyperledger Fabric

The basic transaction mechanism for fabric blockchain is shown in Figure 4.13. There are peer nodes and orderer nodes involved. The blockchain is stored and maintained by peer nodes. There are two types of peer nodes. One type of peer node is called Endorsing Peer, which endorses the transaction proposal submitted to it if the proposal follows the endorsing policies and maintains the status of the blockchain. The other type of peer node is called Committing Peer, which receives the ordered transactions and maintains the status of the blockchain. There is another type of node called Orderer node. It orders all the collected transactions into the right order. A transaction can be completed in four steps. First, the client sends the proposed transaction to the Endorsing Peers that the client connected to. Second, the Endorsing Peers check the proposal against the endorsing policy (i.e., chain code). If succeeded, the Endorsing Peers send the endorsed proposal back to the client. Third, the client sends the endorsed proposal to the orderer nodes, which collects all the transactions from all sources and order them in blocks. Fourth, the orderer nodes send ordered blocks of transactions to the Endorsing Peers and Committing Peers, which accept the transactions and update the world state of the ledger.

The access control was achieved by associating an access control policy with the resource (Hyperledger Fabric, 2018). The policy specifies a rule that evaluates to true or false for a set of identifies. The resource refers to user chaincode, system chaincode, or an events stream source that are used as an endpoint for the user to target at when interacting with Fabric.

The tool used to implement the fabric blockchain is called Hyperledger Composer. The composer is a toolset aims to facilitate the developing of Hyperledger blockchain applications. It supports the Hyperledger Fabric blockchain infrastructure. Figure 4.14 shows the basic file structures that the Hyperledger Composer requires to develop a blockchain application. The figure was made according to the descriptions in the Hyperledger Composer tutorial (The Linux Foundation, 2017). The network is defined by four major files, the Model (.cto), the Script (.js), the Access Control (.acl), and the Query (.qry). After these files are developed, they are combined and exported as an archive file (.bna). Afterward, the archived file is used to deploy the network. The basic model file defines the basic models of the network. It defines three major entities, the participant, the asset, and the transaction. The script file contains the chaincode, which defines the logic of the system, mainly via transaction functions. The access control file defines which participant can do what operations on what resources under what conditions. The query file allows users to define certain SQL like queries.



Created by Yalin Chen (yc.about@gmail.com), 2017

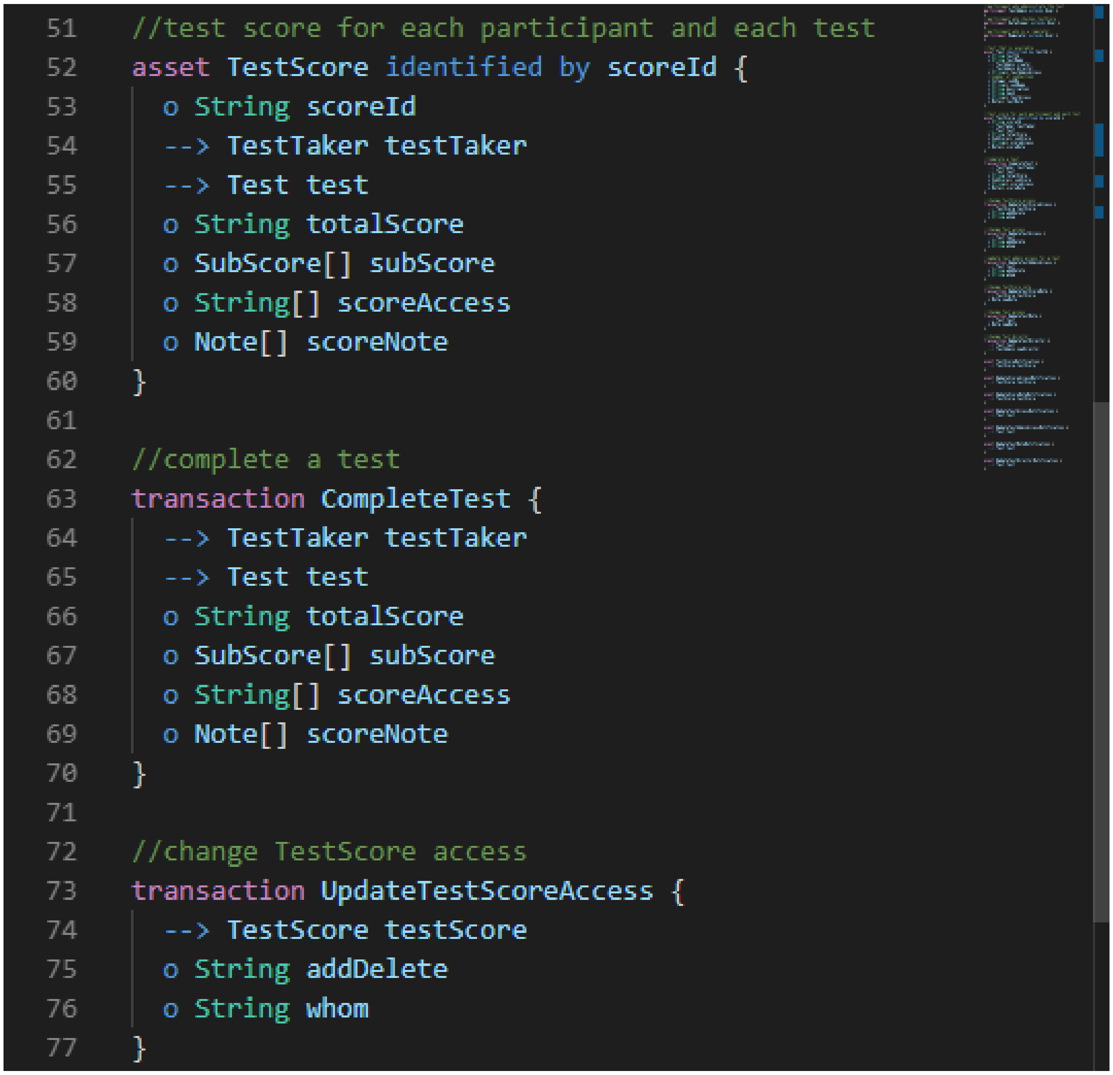
Composer file structure

Implemented by Hyperledger Composer

Figure 4. 14. The basic file structure required by the Hyperledger Composer

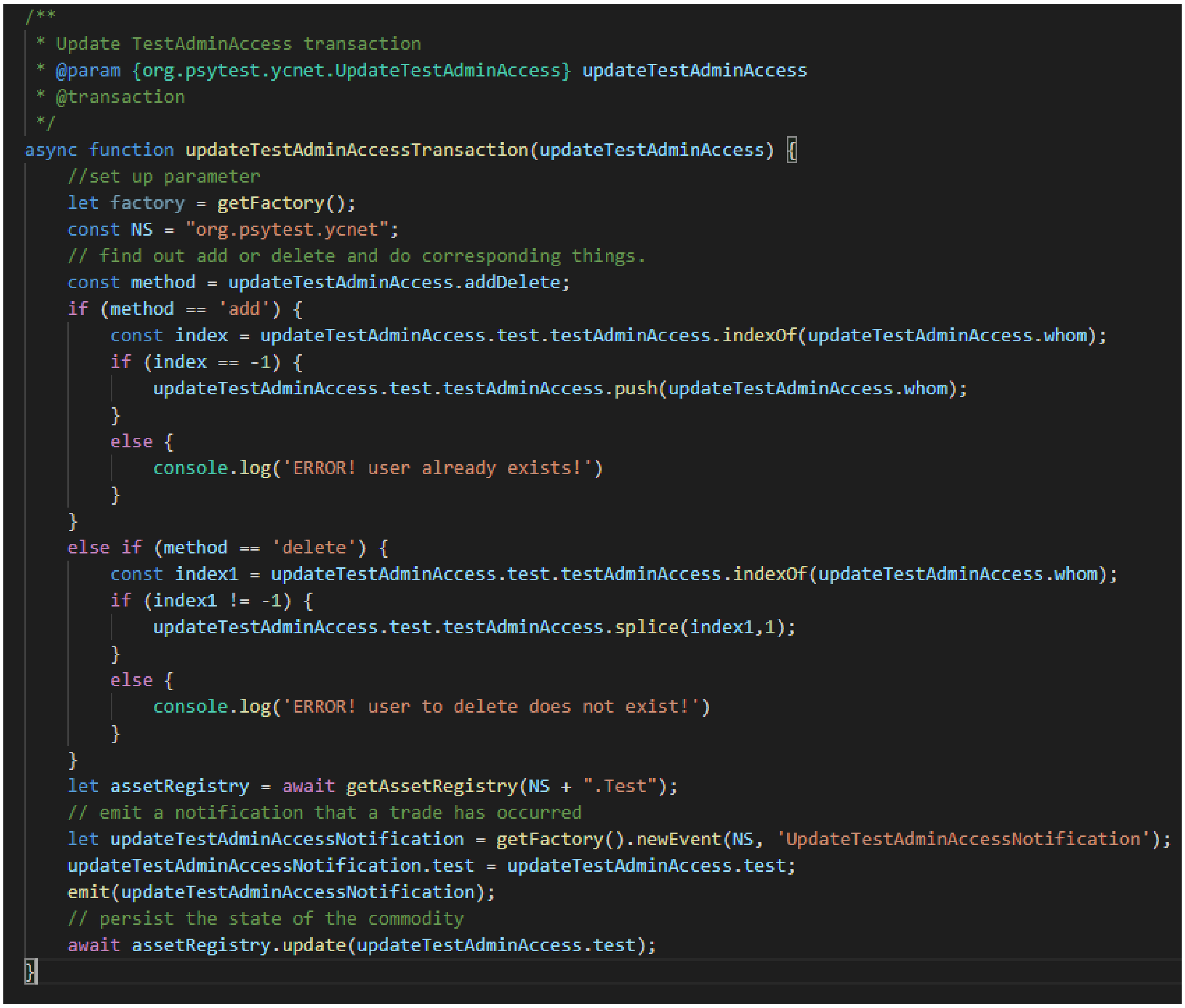
The actual implementation of the blockchain part for the current solution was done by Hyperledger Composer v0.20. The proposed blockchain network elements were created as composer files. The participants, assets, and transactions proposed in the solution structure were created as a model file (.cto) using the Hyperledger Composer Modeling Language. The transaction logic was created as a logic file (.js) using the Javascript language. The access control rules were created as an access control file (.acl) using the Hyperledger Composer Access Control

Language. The following figures (Figure 4.15, 4.16, 4.17) show some code samples for the composer logic file, model file, and permission control file. The network is first tested in the Hyperledger Composer Playground. The Hyperledger Composer Playground is a web interface for quickly developing and testing Hyperledger blockchain. Then, the network is deployed as a Hyperledger Fabric blockchain.



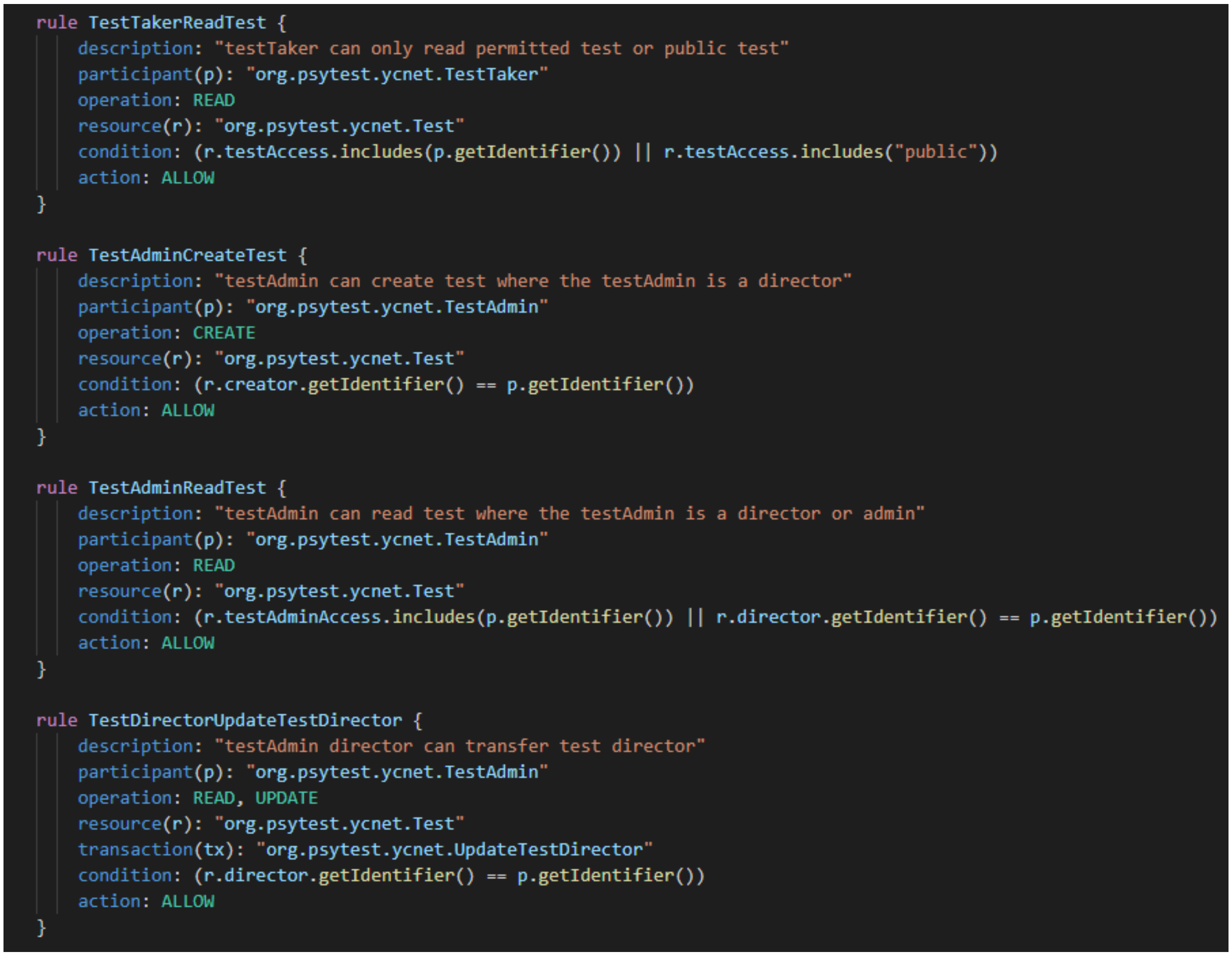
```
//test score for each participant and each test
asset TestScore identified by scoreId {
  o String scoreId
  --> TestTaker testTaker
  --> Test test
  o String totalScore
  o SubScore[] subScore
  o String[] scoreAccess
  o Note[] scoreNote
}

//complete a test
transaction CompleteTest {
  --> TestTaker testTaker
  --> Test test
  o String totalScore
  o SubScore[] subScore
  o String[] scoreAccess
  o Note[] scoreNote
}

//change TestScore access
transaction UpdateTestScoreAccess {
  --> TestScore testScore
  o String addDelete
  o String whom
}
```

Figure 4. 15. Code sample for the model file



```
/**
 * Update TestAdminAccess transaction
 * @param {org.psytest.ycnet.UpdateTestAdminAccess} updateTestAdminAccess
 * @transaction
 */
async function updateTestAdminAccessTransaction(updateTestAdminAccess) {
    //set up parameter
    let factory = getFactory();
    const NS = "org.psytest.ycnet";
    // find out add or delete and do corresponding things.
    const method = updateTestAdminAccess.addDelete;
    if (method == 'add') {
        const index = updateTestAdminAccess.test.testAdminAccess.indexOf(updateTestAdminAccess.whom);
        if (index == -1) {
            updateTestAdminAccess.test.testAdminAccess.push(updateTestAdminAccess.whom);
        }
        else {
            console.log('ERROR! user already exists!')
        }
    }
    else if (method == 'delete') {
        const index1 = updateTestAdminAccess.test.testAdminAccess.indexOf(updateTestAdminAccess.whom);
        if (index1 != -1) {
            updateTestAdminAccess.test.testAdminAccess.splice(index1,1);
        }
        else {
            console.log('ERROR! user to delete does not exist!')
        }
    }
    let assetRegistry = await getAssetRegistry(NS + ".Test");
    // emit a notification that a trade has occurred
    let updateTestAdminAccessNotification = getFactory().newEvent(NS, 'UpdateTestAdminAccessNotification');
    updateTestAdminAccessNotification.test = updateTestAdminAccess.test;
    emit(updateTestAdminAccessNotification);
    // persist the state of the commodity
    await assetRegistry.update(updateTestAdminAccess.test);
}
```

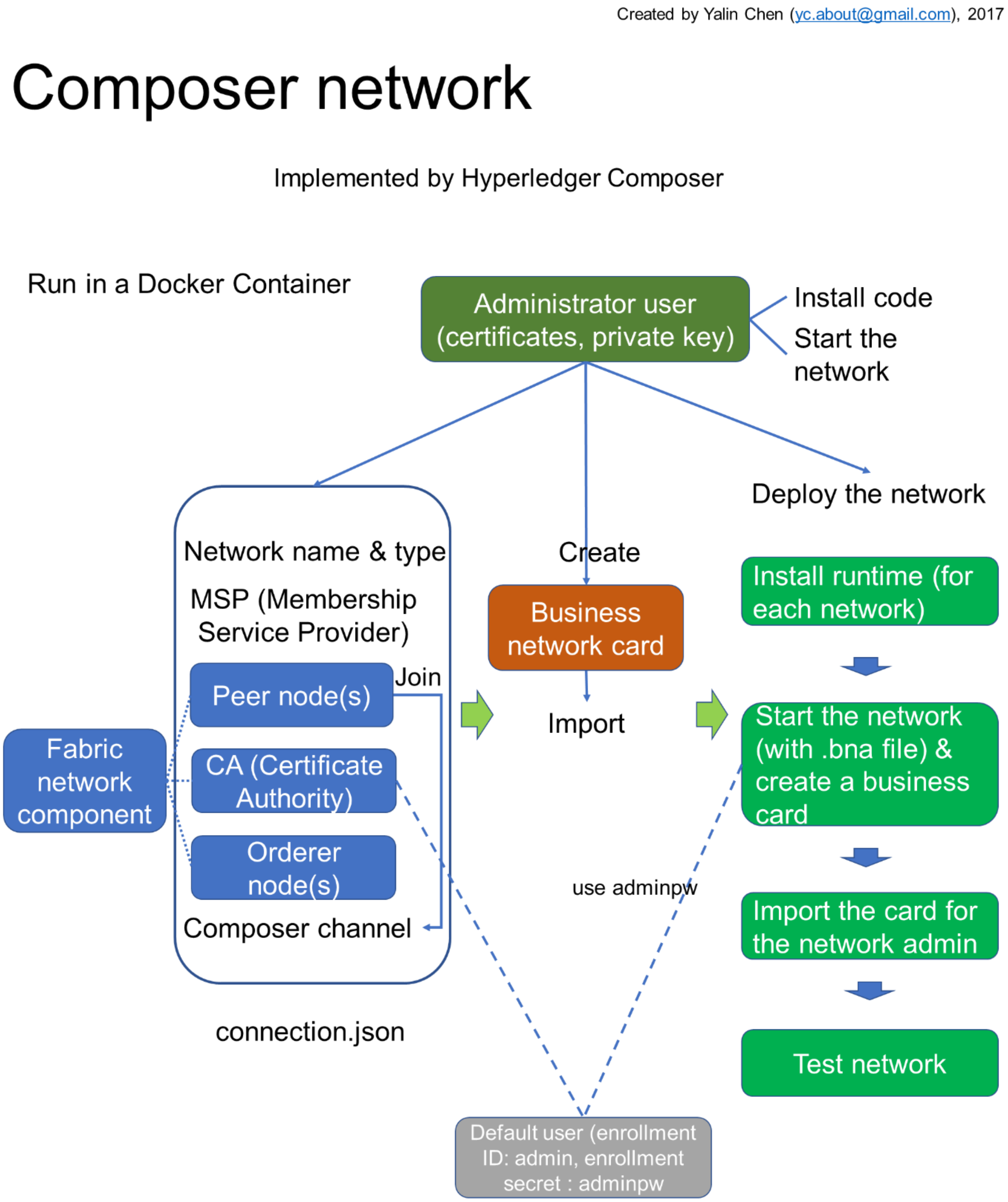
Figure 4. 16. Code sample for the transaction function



```
rule TestTakerReadTest {
    description: "testTaker can only read permitted test or public test"
    participant(p): "org.psytest.ycnet.TestTaker"
    operation: READ
    resource(r): "org.psytest.ycnet.Test"
    condition: (r.testAccess.includes(p.getIdentifier()) || r.testAccess.includes("public"))
    action: ALLOW
}

rule TestAdminCreateTest {
    description: "testAdmin can create test where the testAdmin is a director"
    participant(p): "org.psytest.ycnet.TestAdmin"
    operation: CREATE
    resource(r): "org.psytest.ycnet.Test"
    condition: (r.creator.getIdentifier() == p.getIdentifier())
    action: ALLOW
}

rule TestAdminReadTest {
    description: "testAdmin can read test where the testAdmin is a director or admin"
    participant(p): "org.psytest.ycnet.TestAdmin"
    operation: READ
    resource(r): "org.psytest.ycnet.Test"
    condition: (r.testAdminAccess.includes(p.getIdentifier()) || r.director.getIdentifier() == p.getIdentifier())
    action: ALLOW
}

rule TestDirectorUpdateTestDirector {
    description: "testAdmin director can transfer test director"
    participant(p): "org.psytest.ycnet.TestAdmin"
    operation: READ, UPDATE
    resource(r): "org.psytest.ycnet.Test"
    transaction(tx): "org.psytest.ycnet.UpdateTestDirector"
    condition: (r.director.getIdentifier() == p.getIdentifier())
    action: ALLOW
}
```

Figure 4. 17. Code sample for permission control rules

The basic process for deploying the blockchain network follows the standard Hyperledger Composer deployment procedure. As mentioned above, the composer files were created and combined to a .bna file. Then, the .bna file was deployed according to the procedures described in Figure 4.18. This figure was made according to the descriptions in the Hyperledger Composer tutorial (The Linux Foundation, 2017). The implementation utilizes the Docker tool, make it easier to create, deploy, and run applications. The whole network runs in Docker Containers.



Created by Yalin Chen (yc.about@gmail.com), 2017

Composer network

Implemented by Hyperledger Composer

Run in a Docker Container
Administrator user (certificates, private key)
Install code
Start the network
Deploy the network
Network name & type
MSP (Membership Service Provider)
Peer node(s)
Join
CA (Certificate Authority)
Orderer node(s)
Composer channel
Fabric network component
connection.json
Create
Business network card
Import
use adminpw
Default user (enrollment ID: admin, enrollment secret : adminpw
Install runtime (for each network)
Start the network (with .bna file) & create a business card
Import the card for the network admin
Test network

Figure 4. 18. The basic process of deploying the Hyperledger Fabric blockchain

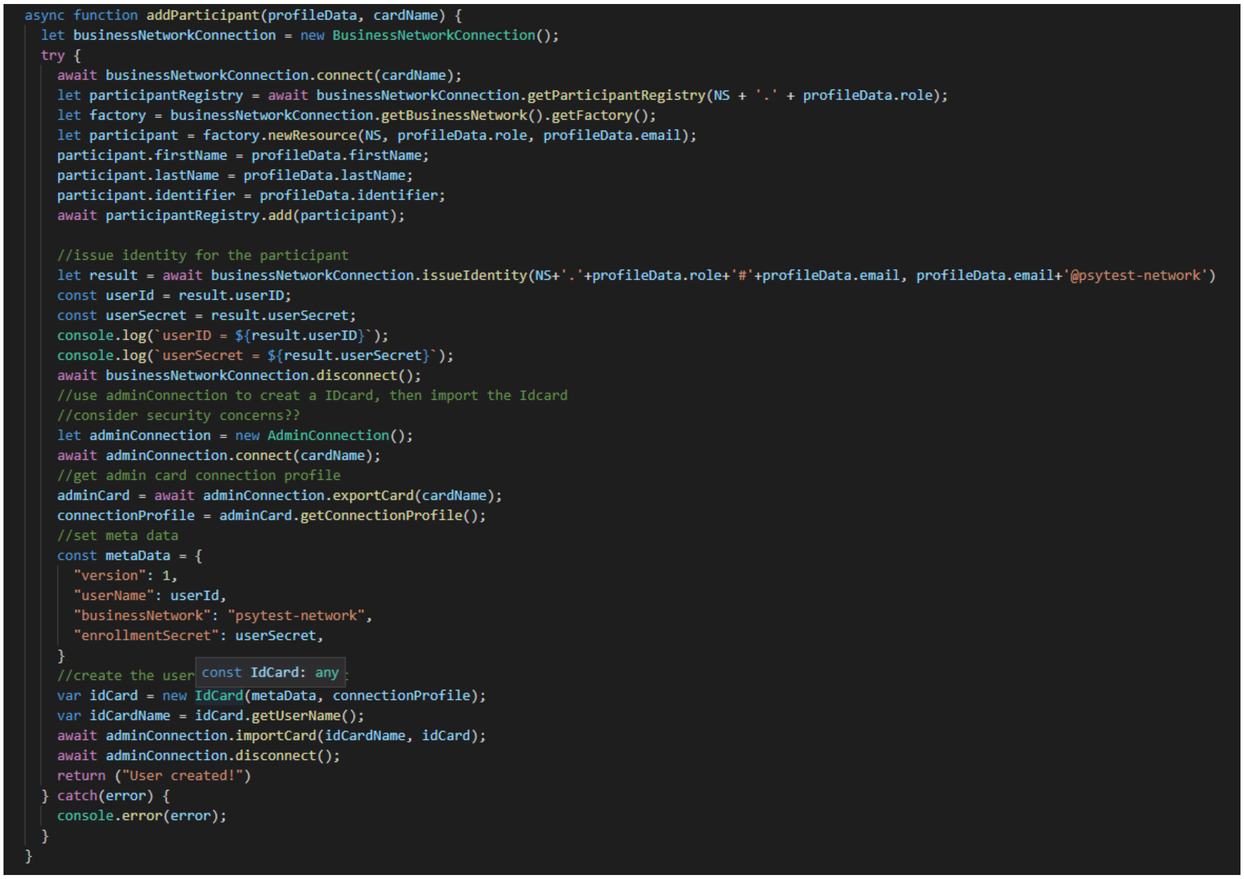
As the first step of the deployment, the peer nodes, certificate authority, and orderer nodes were specified in a connection JSON file. The peer nodes and orderer nodes were mentioned before.

The certificate authority issues certificates in connection with identities. As we mentioned above, the business cards are based on these identities, and different participants interact with the blockchain network based on the different business cards. The peer nodes can join a channel where the information are only shared with peers in the same channel; this can also be specified in the connection JSON file. As the second step, the connection JSON file and the administrator's certificate and private key were combined to create an administrator business card. This administrator user was created in the composer installation process and it installs codes to the network and starts the network. Afterward, the business card was imported to the network and it is used to interact with the network as the ID card of the administrator. As the third step, the archived .bna file was deployed to the network by the administrator user using the card imported. The Hyperledger Composer also provides a REST server component which automatically generates a REST server to interact with the deployed blockchain. However, there is less flexibility, given that all the standard APIs are automatically generated, regardless of needs. Thus, in the current implementation, we did not use the REST server component. Instead, we created a Nodejs web service to directly interact with the deployed blockchain, as will be explained in the following.

Table 4. 4. The current Nodejs web service dependencies

Dependencies	Version number	Description
composer-admin	^0.20.6	Hyperledger Composer administration API
composer-client	^0.20.6	Hyperledger Composer client API
express	^4.16.3	A Web framework for node.
jsonwebtoken	^8.3.0	An implementation of JSON Web Tokens
mongoose	^5.1.4	A MongoDB object modeling tool

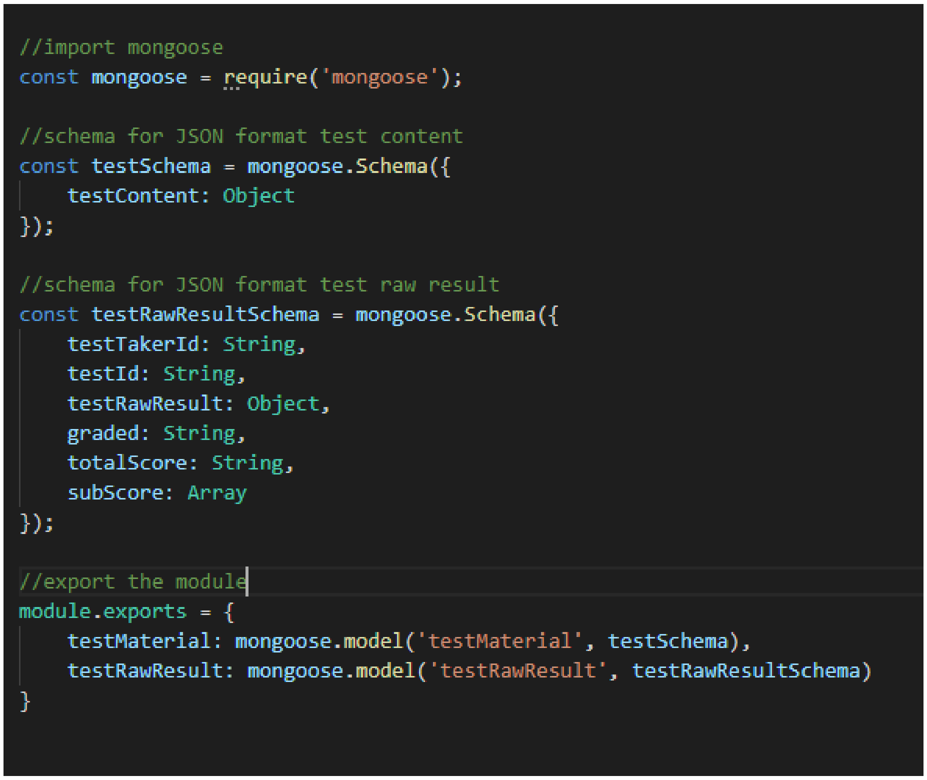
The implementation of the web service used Nodejs with the Expressjs library. The dependencies are shown in Table 4.4. One reason why the current solution chose Nodejs is that the Nodejs applications can directly interact with the blockchain, without the need for an automatically generated REST server. In this way, the developer has better control over what APIs can be provided. As we described before, the web service has a module that handles user registration and login, a module that handles the communication with the no-SQL database, and a module that handles communication with the blockchain.



```
async function addParticipant(profileData, cardName) {
  let businessNetworkConnection = new BusinessNetworkConnection();
  try {
    await businessNetworkConnection.connect(cardName);
    let participantRegistry = await businessNetworkConnection.getParticipantRegistry(NS + '.' + profileData.role);
    let factory = businessNetworkConnection.getBusinessNetwork().getFactory();
    let participant = factory.newResource(NS, profileData.role, profileData.email);
    participant.firstName = profileData.firstName;
    participant.lastName = profileData.lastName;
    participant.identifier = profileData.identifier;
    await participantRegistry.add(participant);

    //issue identity for the participant
    let result = await businessNetworkConnection.issueIdentity(NS+'.'+profileData.role+'#'+profileData.email, profileData.email+'@psytest-network')
    const userId = result.userID;
    const userSecret = result.userSecret;
    console.log(`userID = ${result.userID}`);
    console.log(`userSecret = ${result.userSecret}`);
    await businessNetworkConnection.disconnect();
    //use adminConnection to creat a IDcard, then import the Idcard
    //consider security concerns??
    let adminConnection = new AdminConnection();
    await adminConnection.connect(cardName);
    //get admin card connection profile
    adminCard = await adminConnection.exportCard(cardName);
    connectionProfile = adminCard.getConnectionProfile();
    //set meta data
    const metaData = {
      "version": 1,
      "userName": userId,
      "businessNetwork": "psytest-network",
      "enrollmentSecret": userSecret,
    }
    //create the user const IdCard: any
    var idCard = new IdCard(metaData, connectionProfile);
    var idCardName = idCard.getUserName();
    await adminConnection.importCard(idCardName, idCard);
    await adminConnection.disconnect();
    return ("User created!")
  } catch(error) {
    console.error(error);
  }
}
```

Figure 4. 19. Code sample for adding a participant to the blockchain

For the user registration and login module, different routers and functions were created to receive and handle the user sign up and login requests. For the user sign up, the router receives participant registration information and then pass them to the user registration function. The registration function works in the following ways. First, it encrypts the password using the sha256 algorithm. Then it sends the participant registration information with the encrypted password to the blockchain to create the corresponding participant. Afterward, it requires the blockchain to issue an identity. Then it combines the identity key together with a connection profile and metadata to create a business network card for the current user and import the card to the blockchain network for further use. The sample of the adding participant function is shown in Figure 4.19. After the card has been imported, the function will return the router a success message, which can be sent back as the response to the user's request. For the user log in, the router receives participant login information and then pass them to the user login function. The login function takes the user name, password, and blockchain role and then hash the password using the same sha256 algorithm. Then it compares all the information with the corresponding participant information stored on the blockchain. If all the information matches, a JSON web token (JWT) token will be generated and send back to the router. The router sends the JWT token as the response to the user's login request. The JSON web token for Nodejs library is used to generate the token. To verify the token from the user and to acquire the corresponding card information for the user, a middleware function was created.



```
//import mongoose
const mongoose = require('mongoose');

//schema for JSON format test content
const testSchema = mongoose.Schema({
    testContent: Object
});

//schema for JSON format test raw result
const testRawResultSchema = mongoose.Schema({
    testTakerId: String,
    testId: String,
    testRawResult: Object,
    graded: String,
    totalScore: String,
    subScore: Array
});

//export the module
module.exports = {
    testMaterial: mongoose.model('testMaterial', testSchema),
    testRawResult: mongoose.model('testRawResult', testRawResultSchema)
}
```

Figure 4. 20. Code sample for no-SQL database schema

The MongoDB database was chosen as the database. This is because that the MongoDB is one of the most popular no-SQL databases. For the module that handles communication with the no-SQL database, the Mongoose library was used to help the communication with the MongoDB.

The test materials and raw test results are stored in the MongoDB in the JSON format, as shown in Figure 4.20. All the routers and corresponding functions were developed. For the test material storage, the object id which is automatically generated by the MongoDB after storing the test materials is retrieved and passed as the test ID for the blockchain storage. The sha256 algorithm was used to hash the content of the test materials and the hash value was passed to the blockchain.

For the module that handles communication with the blockchain, all the routers and corresponding functions were developed. As an additional parameter, each function also asks for a business card name, which can be acquired from the JWT token send from the front end. In this way, the access control rules of the blockchain can work properly. The interactions with the blockchain were realized mainly using composer-admin and composer-client node packages.

Table 4. 5. The current front-end dependencies

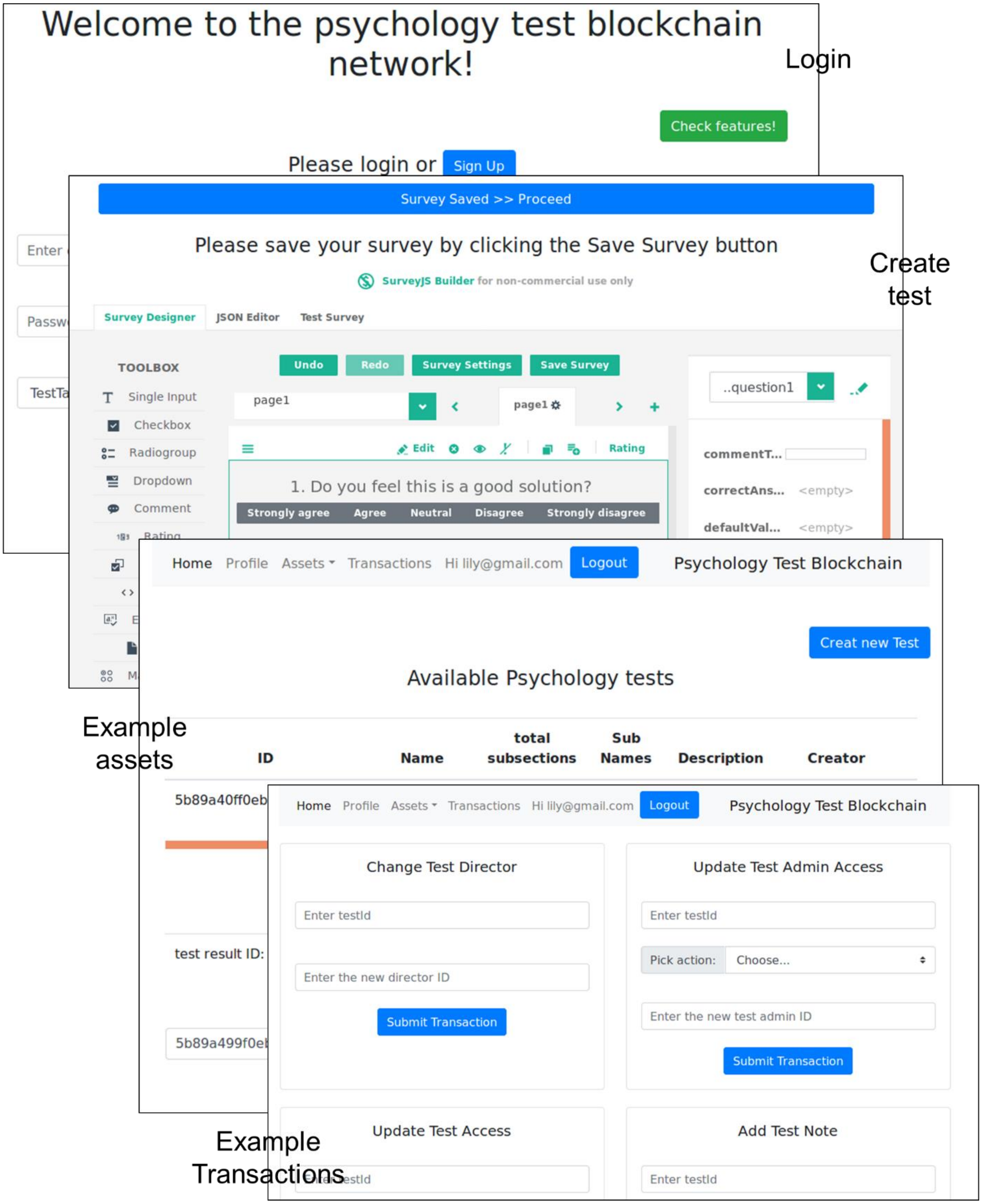
Dependencies	Version number	Description
axios	^0.18.0	Promise based HTTP client
bootstrap	^3.3.7	A Front-end style framework for the usage of surveyjs
jquery	^^3.3.1	A JavaScript library for the usage of surveyjs
jsonwebtoken	^8.3.0	An implementation of JSON Web Tokens
React	^16.4.1	A JavaScript library for building user interfaces.
react-dom	^16.4.1	React package for working with the DOM
react-router-dom	^4.3.1	DOM bindings for React Router
react-scripts	1.1.4	Scripts and configuration used by Create React App
survey-react	^1.0.28	React version of surveyjs to add surveys to the website
surveyjs-editor	^1.0.26	To create or edit JSON for surveyjs library.
surveyjs-widgets	^1.0.26	Custom widgets for the surveyjs library

The implementation of the front end used Reactjs. One reason is that Reactjs is one of the most popular front-end frameworks/libraries. Another reason is that Reactjs if the mobile apps are needed in the future, React Native can easily translate the Reactjs into native IOS or Android apps. The architecture of the front end described above was created. The dependencies are shown in

Table 4.5. On the login page, a link for the user to check the features of the network is provided. To access more functionalities, the user needs to log in. As described in the web service part, when the login information is sent and verified by the web service, a JWT token will be issued to the front end. This JWT token is saved in the local storage and expires after a certain amount of time. The participant role information contained in the JWT token is decoded using the JSON web token node package and then the relevant pages will be rendered to the participant.

To handle the creation, display, and completion of the test and view test result, the Surveyjs Javascript library is used. It is an open-source library for survey development and usage under MIT license. The front-end embedded functionalities provided by the Surveyjs. In addition, it also embedded the survey builder functionalities provided by the Surveyjs. In this way, the user can easily build questionnaires within the current front end. It is worth to note that although the Surveyjs library is free with no conditions, the embedded version of the survey builder is only free for academic use. For commercial use, a license should be purchased. As an alternative, the survey builder can be used through the original provider's website.

Figure 4.21 shows some example pages of the front end. Four pages are shown including user login, questionnaire creation, test asset status, and transaction submission. For the user login, the user needs to provide a user name, password, and participant role. The test administrator can create new test via the test builder on the create test page. The test administrator (e.g., lily@gmail.com) can check the available test assets and score the available raw results. The test administrator can also submit transactions through the example transaction page. The login out button is shown at the navigation bar. There are many other pages were created, but they were not included due to the word limit.



Welcome to the psychology test blockchain network!

Please login or Sign Up

Please save your survey by clicking the Save Survey button

1. Do you feel this is a good solution?

Available Psychology tests

ID	Name	total subsections	Sub Names	Description	Creator
5b89a40ff0eb					

test result ID:

5b89a499f0eb

Figure 4. 21. Example front-end pages

CHAPTER 5 PERFORMANCE EVALUATION

In answering the third research question, which asked for the performance of the current solution, an experiment was designed and carried out in the cloud environment to assess the performance of the current system. Although there is a no-SQL database, the focus here is on the performance of blockchain. This is because that blockchain is the performance bottleneck for the current solution.

For the experiment, latency is used as the major dependent variables. RT refers to when sending a request to the server, how long to get a response after it has communicated with the blockchain. Latency was chosen because latency usually is a bottleneck of blockchain applications.

There are two independent variables, request type and number of clients. First, there are two major HTTP methods related to the current application, they are Get and Post. Different methods may have different performances. More specifically, the Get method should be faster because it is acquiring information from the blockchain, which does not change the world state of the common ledger. The Post method should be slower because it requires to write information to the blockchain, which will change the world state of the common ledger. Thus, the type of method was selected as an independent variable. Second, the number of clients that are making the requests can affect the performance. As the number of clients increases, the performance may decrease. Thus, the number of clients was selected as another independent variable. Note that number of nodes was not taken as a directly manipulated variable for several reasons. First, the current solution is not focusing on the exact performance of the Hyperledger Fabric, but the basic performance of the current solution within one implementation (Hyperledger Fabric via Composer). Second, the multiple nodes mode of the Hyperledger Fabric via Composer is not mature enough (i.e., no detailed enough official documents are provided at the time, only a simple demonstration). Third, if one is interested, the latency for multiple nodes can be estimated based on the data collected on one node. For instance, the time it takes for the blockchain system to write information to blockchain and change the common ledger status on one peer node can be acquired by subtracting the Get request latency from the Post request latency. Thus, the time acquired can be used as the estimation of the time to write information to the blockchain for an additional node. This is because that each node function similarly. For the Get request, because data can be acquired from one node, there should not be a significant difference between different numbers of nodes.

The design of this study is 2 (HTTP methods) x 6 (number of clients) and is summarized in Table 5.1. As mentioned above, the 2 levels of HTTP methods are Get and Post. The number of clients was divided into 6 levels. Those are 1 client, 50 clients, 100 clients, 150 clients, 200 clients, and 400 clients. The six levels here help to determine whether the performance drops linearly or not as the number of clients increases. The performance under 1 client condition provides the baseline performance for the solution.

Table 5. 1. Summary of experiment variables

Independent Variable		Dependent Variable
Request type	Number of Clients	
Get request × 2	1 client (baseline)	Response latency
	50 clients	
	100 clients	
	150 clients	
	200 clients	
	400 clients	
Post request × 2	1 client (baseline)	
	50 clients	
	100 clients	
	150 clients	
	200 clients	
	400 clients	

Two predictions can be made. The first prediction is that the Get method will have better performance compared to the Post method, given that the Post method requires an actual change in the common ledger. The second prediction is that as the number of clients grows linearly, the performance will drop linearly, given that each additional client needs to take a certain amount of time and resources.

5.1 Method

5.1.1 Instruments

A windows computer was used as the client simulator. The specification of the computer is shown in Table 5.2. JMeter 5.0 (The Apache Software Foundation, 2019) was used as the main performance evaluation tool. It is an open source 100% Java application software designed to test functional behavior and measure performance. It can be used to simulate a heavy load on a server to test its strength or to analyze the performance.

Table 5. 2. The client computer specifications

Operating System	Windows 10 Home
Computer Model:	Lenovo ThinkPad T470 Signature Edition
Processor:	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71GHz
Memory	16.0GB (15.9GB usable)
System type	64-bit Operating System, x64-based processor

The server was deployed to GOOGL cloud using GOOGL Compute Engine. The specification of the cloud machine is shown in Table 5.3. For the Get and Post methods, four representative API paths were chosen among the major API paths to measure the performance. The get participant and get test API paths were chosen for the Get method. The complete test and update test access API paths were chosen for the Post method. The user authorization part was set to always true for the performance evaluation purpose.

Table 5. 3. The cloud server specifications

Operating System	Ubuntu 16.04 LTS
Machine type:	n1-highmem-2 (2 vCPUs, 13 GB memory)
CPU platform:	Intel Haswell (Intel Xeon E5 v3 @2.3GHz)
Zone	us-east1-b

Four hundred experimental participants with one hundred each for test taker, test admin, data viewer, and regulator were created as experimental participants. Four hundred experimental tests were created as experimental tests. The complete test transaction takes in one test taker and one test and produces a test score. The update test access transaction takes in one test and adds a user to the test access field.

5.1.2 Procedure

The test plan was set up in the GUI mode of the JMeter, and the test was run in the non-GUI mode, as suggested by the JMeter Manual. First, the server was set to run on Google Cloud. Then the 400 sample participants and sample tests were created. Afterward, the 1 client condition was administrated for the two Get methods and two Post methods. Then the 50-client condition was administrated for the two Get methods and two Post methods, each of the clients targeted one unique resource under each request. Afterward, the 100-client, 150-client, 200-client, and 400-client conditions were administrated by specifying the corresponding number of users for the two Get requests and two Post requests. As in the 50-client condition, each of the clients targeted one unique resource under each request. For all the measurements, the ramp-up period was set to 1s.

Given that the main purpose of the experiment is to provide an approximate performance evaluation and two measurements of each request type under each client condition were used, thus one round rather than multiple rounds of the test was administrated. In addition, to estimate the difference between the Get request and the Post request, the measurements under six levels of number of clients can be taken as six measurements of the difference to calculate average, the standard deviation (SD), the 95% confidence intervals (CI), and the 99% CI for the difference.

5.2 Results

First, the overall performance for the two Get and Post requests was calculated. It provides an overall picture of the current solution. Afterward, the result was broken down into each individual request level under each request type. In this way, detailed performance can be demonstrated and compared. Finally, the estimation for the blockchain system to write information and update the common ledger status was calculated.

Overall, for the Get method and Post method, data were averaged over each of the two requests for each number of client condition, respectively. This provides a more accurate measurement of the overall performance compared to a measurement of a single request. The overall performance is shown in Figure 5.1. First, the baseline performance can be seen under the 1 client condition. The latency for the Get request is 496 ms whereas for the Post request is 3305 ms. The relative slower Post request compare to the Get request can be seen across the whole range of the number of clients tested. This confirmed the prediction that the Post request is slower compared to the Get request. The reason is that the Post request requires new information to be written to the blockchain and update the common ledger status, which takes extra time. Second, as the number of clients increases, the latency increases linearly. For the Get request, there is a nearly perfect linear increase in latency as the number of clients increases, where the latency increased from 496 ms to 82355 ms. The Post request showed a similar trend, where the latency increased from 3305 ms to 88535 ms. Thus, both the Get request and the Post request confirmed the second prediction, which predicted that as the number of clients grows linearly, the performance will drop (i.e., increase in response latency) linearly given that each additional client needs to take a certain amount of time and resources.

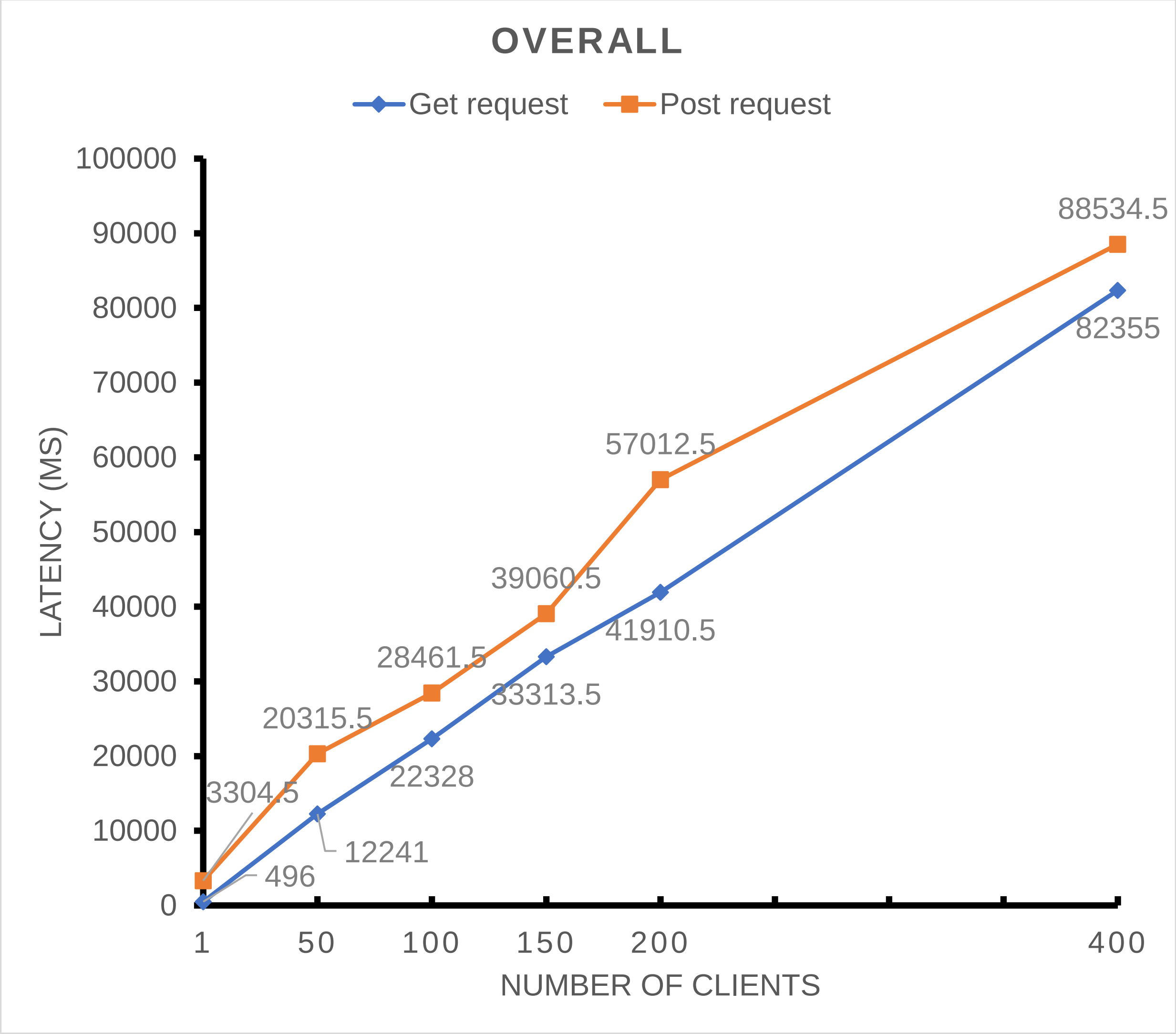


Figure 5. 1. The overall performance by number of clients and request type

The performance of the two Get requests is shown in Figure 5.2. The result is relatively straightforward. The baseline latency for 1 client condition is 541 ms and 451 ms for the two requests. As the client number increases, the latency for Get participant and Get test increased in a perfectly linear manner. At the 400-client level, the latency reaches 85561 and 79149 ms, respectively. The two requests look identical to each other across the levels of the number of clients, although the get participant request is arguably slower than the get test request.

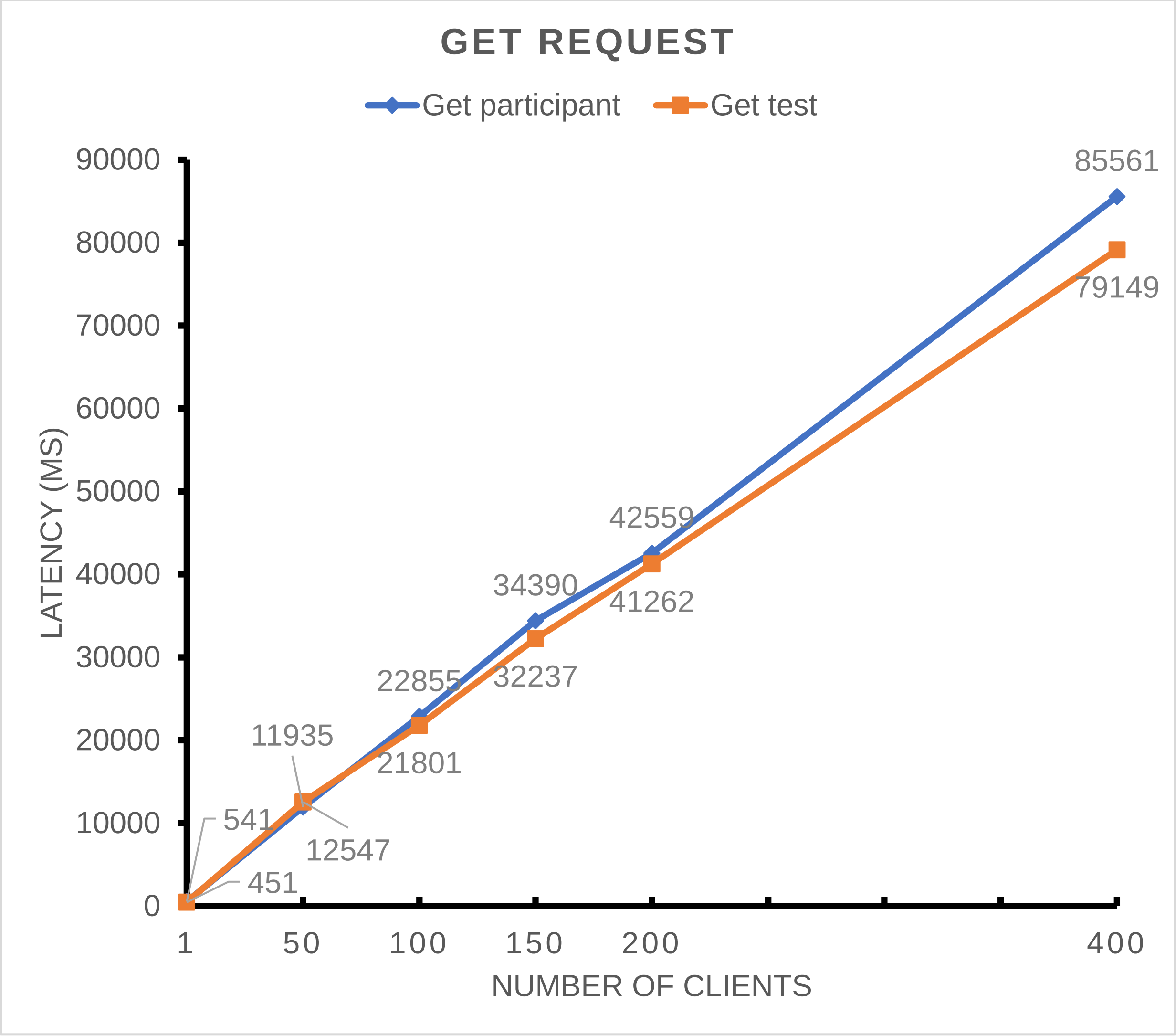


Figure 5. 2. The latency performance for the Get request

The performance for the two Post requests is shown in Figure 5.3. The basic latency for 1 client condition is 2557 ms and 4052 ms for the two requests. For the Post request, the latency performance for the two requests also seems similar. For the complete test transaction, the latency grows linearly as the number of clients increases. The case for the update test access transaction is similar, although a bit faster.

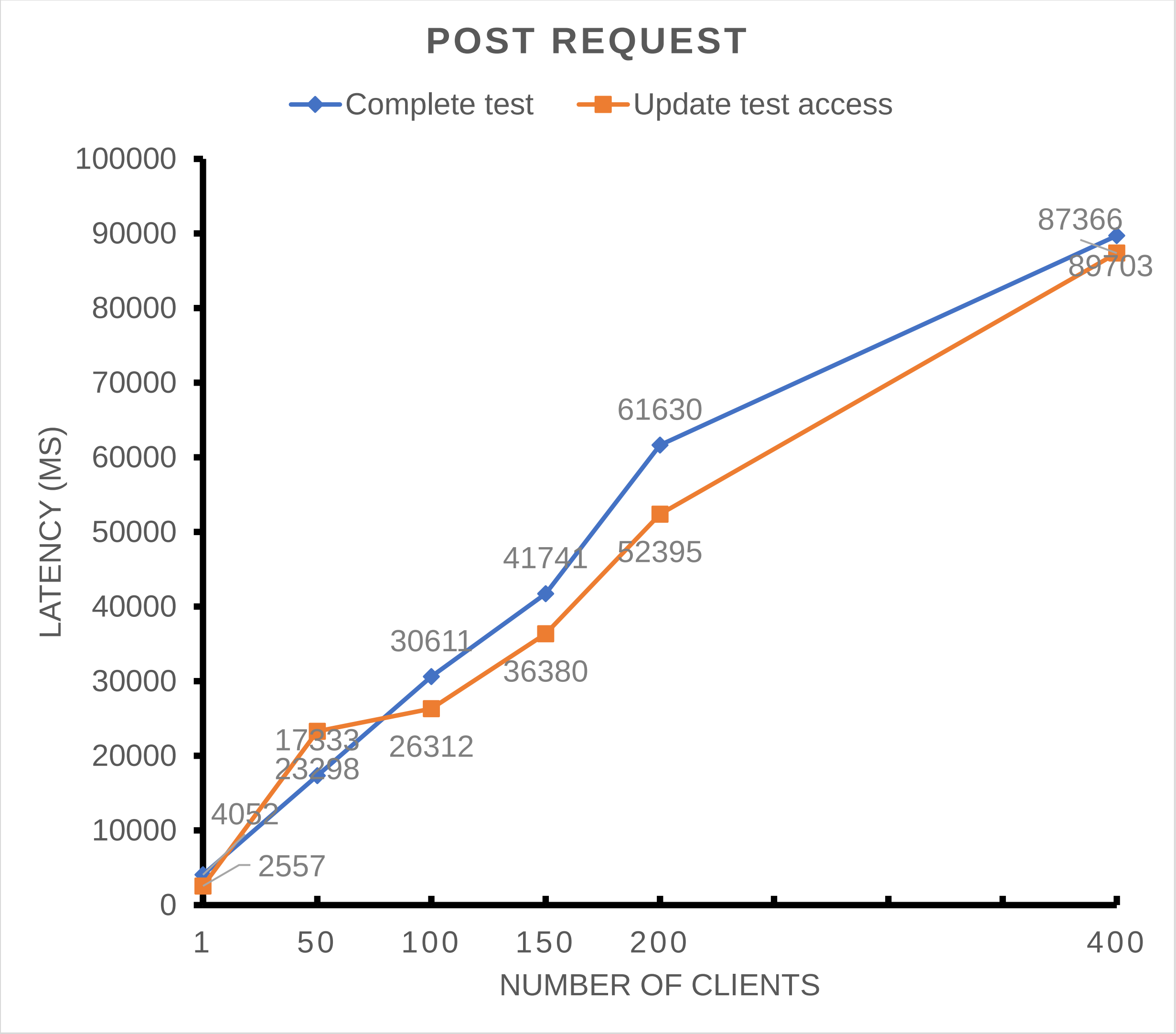


Figure 5. 3. The latency performance for the Post request

In addition, the number of nodes effect was estimated. The overall difference between the Post request and the Get request is shown in Figure 5.4. The figure was acquired by subtracting the overall Get request latency from the overall Post request latency. The overall average difference across all the number of clients is 7341 ms with the SD of 4163 ms. The 95% CI (±1.96SE) calculated is 4010 – 10672 ms. The 99% CI (±2.58SE) calculated is 2963 – 11719 ms. This is the estimation for the time it takes for the blockchain system to write information to blockchain and change the common ledger status on one peer node. Thus, if an additional node is added, it functions similarly as this node, given that different nodes are running in parallel and independently. As a result, there will not be a significant increase in latency.

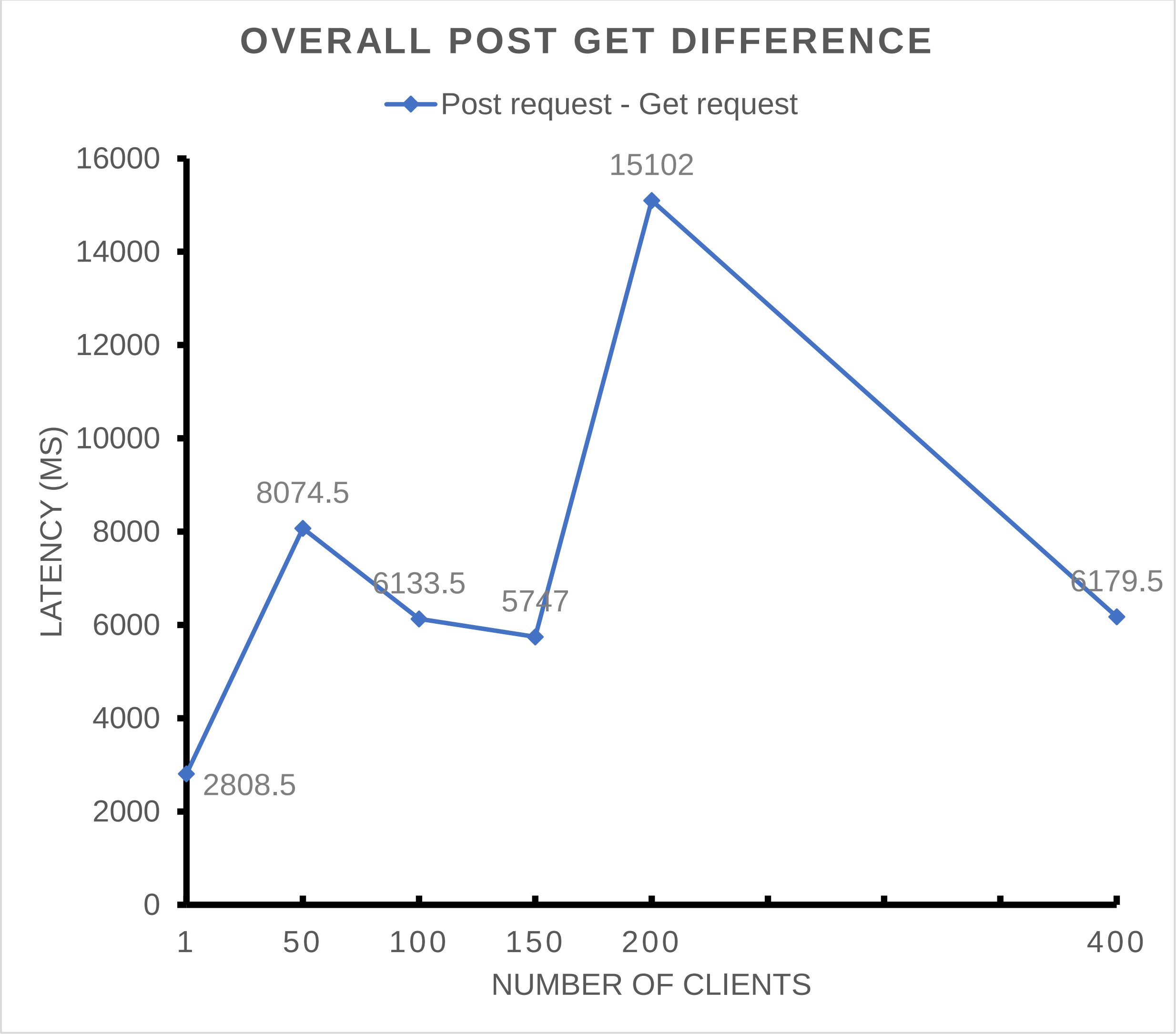


Figure 5. 4. The overall difference between the Post request and the Get request

5.3 Discussion

In the experiment, 2 levels of request methods and 6 levels of number of clients were manipulated and the performance was measured. The 2 levels of HTTP methods are Get and Post. The numbers of clients are 1 client, 50 clients, 100 clients, 150 clients, 200 clients, and 400 clients.

Two predictions were made. First, the Get method will have better performance compared to the Post method. Second, as the number of clients grows linearly, the performance will drop linearly. The results confirmed both of our predictions. Specifically, first, the Post request is 2809 ms slower than the Get request under 1 client condition and 7341 ms slower on average across the range tested. Second, as the number of clients goes linearly from 1 to 400, the latency of both the Post request and the Get request grows linearly from 3305 ms to 88535 ms and from 496 ms to 82355 ms respectively.

The average 7341 ms difference between the Post request and the Get request reflected the time it takes for the blockchain system to write information to blockchain and change the common ledger status on one peer node. If an additional node is added, because different nodes are running in parallel and independently, there will not be a significant increase in the latency.

The latencies acquired are relatively good. With the number of clients increased, the latency increases. However, with 200 clients making requests at the same time, the performance is still below 1 minute and with 400 clients, the performance is still below 2 minutes. The 200 clients per second translate to 72,000 clients per hour and the 400 clients per second translate to 144,000 clients per hour. This is adequate for a psychology test network given that people do not take psychology tests frequently. The average latency for the blockchain system to write information to blockchain and change the common ledger status is less than 8 seconds within the range of clients tested, and the estimated ceiling of 99% CI is under 12 seconds. This is significantly faster compared to the traditional blockchain systems. For instance, it takes about 10 minutes for the bitcoin system to finish a transaction.

CHAPTER 6 DISCUSSION

Distributed ledger is a new decentralized technology in data management. It represents one of the newest and most exciting trends in moving from a human-scaled society to a computer or machine scaled society. This thesis explores the possibility of adopting the new distributed ledger technology, represented by blockchain, to psychology test data management.

In this thesis, the relevant literature was reviewed and the benefits of using blockchain for psychology test data management were identified. A complete academic solution was proposed with the detailed architecture and its implementation. The performance was evaluated in an experiment and the results were discussed.

6.1 Research questions

In this thesis, three research questions were proposed. They include identifying the benefits of using the distributed ledger technology in psychology test data management, designing and implementing a theoretical structure of the distributed ledger solution, and evaluating the performance of the solution.

In answering the first research question, which asked for the benefits of using blockchain to manage psychology test data, the relevant literature was reviewed, and the benefits were summarized. There are three major benefits in applying the distributed ledger technology to psychology test data management. First, using distributed ledger provides a better way to establish and maintain a single source of truth for psychology test data. The distributed ledger technology removes the need for a central party in transactions but guarantees the security and successfulness of the transactions. It usually contains a common ledger that each node within the network has the exact same copy. The ledger contains multiple chained blocks. Except the first and the last block, each block is connected to a previous block and each block is connected by a subsequent block. Each block stores logs of the event (i.e., transactions). The log of events is done by a computer network. Within the network, each node verifies each transaction and keeps a copy of a ledger, which represents a single truth of history. This structure makes the distributed ledger immutable. In this way, untrusted parties can make transactions securely. In other words, the distributed ledger technology represented by blockchain removes the need of the trust (i.e., to the central node) to build a single truth of history. One major obstacle in building a single source of truth for psychology test data management using traditionally centralized database is the lack of trust to the central node. With blockchain, the single source of truth can be built without trust.

Second, the distributed ledger technology makes psychology test related data immutable and available everywhere. Due to the nature of the blockchain, the data are immutable and replicated in every node. This makes psychology test data and access control data secure and available on every node. The Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017) requires psychologists to carry certain duties to protect the data. For instance, test materials need to be well protected to prevent the misuses, test result data need to be protected for patients' or clients' privacy. This requires the access control of the data to be highly secure. In addition, psychology test data should be highly available to facilitate usage in practice. This requires that psychology test data and access control data be highly available. The blockchain can make the data more secure by making it immutable and can make the data highly available by replicating it on every node.

Third, the distributed ledger technology grants people the power to control their own data in a distributed way. In a psychology test, the data collected usually are highly sensitive and about the subject. The Ethical Principle of Psychologist and Code of Conduct (American Psychology Association, 2017) requires that in disclosing confidential information, there should be appropriate consent of the organizational client, the individual client/patient, or another legally authorized person on behalf of the client/patient unless prohibited by law. In practice, however, the information is usually stored in a centralized database where many different parties have access to the database. The database administrators can grant access whereas the patients or clients do not have or have very little control. Without a central party controlling everything, the blockchain technology can grant people the real power in controlling their own data.

In answering the second research question, which asked for an academic solution, a complete academic solution using the blockchain technology for psychology test data management was proposed and implemented. The solution includes a permissioned blockchain, a centralized database, a web service, and a front end. The solution stores part of the data on the blockchain and the rest of the data into a centralized database off the blockchain. Because all the data need to be replicated in every node of the blockchain, the more data stored on the blockchain, the slower the overall performance will be. Also, to utilize the immutability and the availability features of the blockchain, important data need to be stored on the blockchain. Thus, as a balanced solution, the current solution stores final test scores, test metadata, and access control data on the blockchain, whereas the test materials and raw test results off the blockchain into a centralized database. Meanwhile, to make the centralized database uses the immutability feature of the blockchain, the hash values of the materials that are stored in the centralized database are also stored on the blockchain. The web service component mainly connects the user from the front end, the blockchain, and the non-blockchain database. It receives user requests from certain APIs. Depending on the requests, it communicates with the blockchain, non-blockchain database, or both to provide proper responses to the user requests. The web service contains three basic modules: a module that handles user registration and authorization, a module that handles communication with the centralized database, and a module that handles communication with the blockchain. The front end defines the interface for the user to communicate with the blockchain and the centralized database via the web service. The front end collects the user inputs and sends them to the web service via different APIs that were designed. The front end also receives the responses and displays them properly to the end users. Depending on the role that a user is acting as, different interfaces are provided. The front end involves a login component to authorize the user, a sign-up component if the user does not yet exist in the network as a participant, and a third component that involves available functions depending on the role of the user.

In answering the third research question which asked for the performance of the current solution, the performance of the proposed solution was evaluated. Two levels of request method and 6 levels of client number were manipulated. The 2 levels of HTTP methods are Get and Post. The numbers of clients are 1 client, 50 clients, 100 clients, 150 clients, 200 clients, and 400 clients. The results show that the Post request is slower than the Get request and as the number of clients grows linearly, the latency of the requests grows linearly. The average 7341 ms (SD = 4163 ms, 95% CI [4010 – 10672 ms]) slower latency for the Post request compared to the Get request reflected the time it takes for the blockchain system to write information to blockchain and change the common ledger status on one peer node. If an additional node is added, because that different nodes are running in parallel and independently, there will not be a significant increase in the latency.

6.2 Conclusion

Psychology tests are widely used in mental health diagnoses, education assessments, and recruitment assessments. Several major problems exist in using the traditional way to manage psychology test data. First, there is a lack of a single source of truth for psychology test data due to centralized storage. Second, the data are mutable and have the risk of a single point of failure. Third, people have very weak or no control of their own data. Blockchain technology can bring major benefits to psychology test data management. Three major benefits are that blockchain provides a better way to establish and maintain a single source of truth for management of psychology test data, it makes psychology test data and access control data immutable and highly available, and it grants people the power in controlling their own data in a distributed way. A complete academic solution was proposed. The solution includes a permissioned blockchain, a no-SQL database, a web service, and a front-end. The blockchain stores user data, metadata of psychology tests, final test scores, and access control data of the tests and test scores. The blockchain also stores all the transaction logs. The no-SQL database stores test materials and raw test results. The web service interacts with the blockchain and the no-SQL database, whereas the front-end interacts with the web service. The performance was evaluated under different types of requests and numbers of clients. Results showed that the Post request is slower than the Get request, and as the number of clients increases, the latency increases. The average 7341 ms (SD = 4163 ms, 95% CI [4010 – 10672 ms]) slower latency for the Post request compared to the Get request reflected the time for the blockchain system to write information in and change the common ledger status on one peer node. The solution proposed here provides a new way to manage psychology test data with satisfactory performance.

6.3 Future directions

The current thesis provides a potential academic solution for psychology test data management using the distributed ledger technology. There are several questions waiting for future research to address.

First, future research can focus on extending the current solution to other questionnaire data management. As mentioned above, although the solution here focuses on psychology test data management, it may also be applied to other areas that involve sensitive questionnaire-based data.

Second, the management of other forms of psychology assessment data using blockchain needs to be explored. Current thesis designed a solution for questionnaire-based psychology assessment data management, but there are other forms of psychology assessment data such as data of physiological assessments or neuropsychological assessments. How to manage those data through blockchain can be another future research direction.

Lastly, the current solution needs to be optimized. Here, this thesis focuses on providing an academic solution. There are improvements can be done. For instance, is it necessary to have a participant name stored on blockchain? If no name is stored on the blockchain, how does one connect the test results with certain individuals? Are there better ways to design the web service functions that can boost the performance? Moreover, the user experience for the front-end can be improved. In addition, the deletion of psychology test data needs to be explored. More specifically, data cannot be deleted after they are produced because they are stored on the blockchain. If for some reason the data need to be deleted, how to handle that could be another future research direction.

REFERENCE

Angeletti, F., Chatzigiannakis, I., & Vitaletti, A. (2017). Privacy preserving data management in recruiting participants for digital clinical trials. *Proceedings of the First International Workshop on Human-centered Sensing, Networking, and Systems (HumanSys'17), ACM, New York, NY, USA*, 7-12. DOI: https://doi.org/10.1145/3144730.3144733

American Psychology Association. *Ethical principles of psychologists and code of conduct*. Retrieved February 20, 2018 from http://www.apa.org/ethics/code/ethics-code-2017.pdf

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using Blockchain for Medical Data Access and Permission Management. *2016 2nd International Conference on Open and Big Data (OBD), Vienna*, 25-30. doi: 10.1109/OBD.2016.11

Bauer, R. M., Iverson, G. L., Cernich, A. N., Binder, L. M., Ruff, R. M., & Naugle, R. I. (2012). Computerized neuropsychological assessment devices: Joint position paper of the American Academy of Clinical Neuropsychology and the National Academy of Neuropsychology. *Archives of Clinical Neuropsychology, 27*, 362–373. doi:10.1093/arclin/acs027

Baxendale G. (2016). Can Blockchain Revolutionise EPRs? *ITNOW, 58*(1), 38–39.

Beninger, P. & Ibara, M. A. (2016). Pharmacovigilance and biomedical informatics: a model for future development. *Clinical Therapeutics, 38*(12), 2514-2525.

Bilder, R. (2011). Neuropsychology 3.0: Evidence-based science and practice. *Journal of the International Neuropsychological Society*, *17*, 7–13. doi:10.1017/S1355617710001396

Butcher. J. N. (2003). Computerized psychological assessment. In J. R. Graham, & J. A. Naglieri (Eds.), *Handbook of psychology: assessment psychology (pp. 141-163)*. New York: John Wiley & Sons.

Butcher, J. N., Perry, J., & Hahn, J. (2004). Computers in clinical assessment: historical developments, present status, and future challenges. *Journal of Clinical Psychology*, *60*, 331–345.

Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 1-14.

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. *IEEE Access*, *4*, 2292–2303.

Crosby, M., Nachiappan, Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. *Applied Innovation Review, 2*, 6–10.

Dede, E., Zalonis, I., Gatzonis, S., Sakas, D. (2015). Integration of computers in cognitive assessment and level of comprehensiveness of frequently used computerized batteries. *Neurology, Psychiatry and Brain Research, 21*(3), 128-135.

Dubovitskaya, A., Xu, Z., Ryu, S., Schumacher, M., & Wang, F. (2017). Secure and Trustable Electronic Medical Records Sharing using Blockchain. *arXiv preprint arXiv:1709.06528*.

Ethereum White Paper (2018). *A Next-Generation Smart Contract and Decentralized Application Platform*. Retrieved February 26, 2018, from https://github.com/ethereum/wiki/wiki/White-Paper#decentralized-autonomous-organizations

Google. (2018). *The Go Programming Language*. Retrieved March 1, 2018, from https://golang.org/doc/

Greenspan, G. (2015). *Ending the Bitcoin vs Blockchain debate.* Retrieved February 12, 2018, from https://www.multichain.com/blog/2015/07/bitcoin-vs-blockchain-debate/.

Groth-Marnat, G. (1999). Financial efficacy of clinical assessment: Rational guidelines and issues for future research. *Journal of Clinical Psychology, 55*, 813–824.

Groth-Marnat, G. (2002a). Introduction. In *Handbook of psychological assessment (pp.1-36)*. Hoboken: John Wiley & Sons, Inc.

Groth-Marnat, G. (2002b). Context of Clinical Assessment. In *Handbook of psychological assessment (pp.37-68)*. Hoboken: John Wiley & Sons, Inc.

Groth-Marnat, G., & Edkins, G. (1996). Professional psychologists in general health care settings: A review of the financial efficacy of direct treatment interventions. *Professional Psychology: Research and Practice, 27*, 161–174.

Hyperledger Fabric. (2018). Retrieved March 13, 2018, from http://hyperledger-fabric.readthedocs.io/en/latest/arch-deep-dive.html

Kish, L. J., & Topol, E. J. (2015). Unpatients– why patients should own their medical data. *Nature biotechnology, 33*(9): 921–924. doi:10.1038/nbt.3340

Kuo, T. T., Kim, H. E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and health care applications. *Journal of the American Medical Informatics Association, 24*(6), 1211-1220.

Liu, W., Zhu, S. S., Mundie, T., & Krieger, U. (2017). Advanced block-chain architecture for e-health systems. *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian,* 1-6. doi: 10.1109/HealthCom.2017.8210847

McMinn, M. R., Buchanan, T., Ellens, B. M., & Ryan, M. K. (1999). Technology, professional practice, and ethics: Survey findings and implications. *Professional Psychology: Research and Practice, 30*, 165–172.

McMinn, M. R., Ellens, B. M., & Soref, E. (1999). Ethical perspectives and practice behaviors involving computer-based test interpretation. *Assessment, 6*, 71–77.

Mettler, M. (2016). Blockchain technology in healthcare: The revolution starts here. In e-Health Networking, Applications and Services (Healthcom), *2016 IEEE 18th International Conference. IEEE*, 1-3.

Mukhopadhyay, U., Skjellum, A., Hambolu, O., Oakley, J., Yu, L., & Brooks, R. (2016). A brief survey of cryptocurrency systems. *2016 14th Annual Conference on Privacy, Security and Trust (PST). IEEE*, 745–752. doi: 10.1109/PST.2016.7906988

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* Retrieved February 12, 2018, from https://bitcoin.org/bitcoin.pdf.

Nichol, P. B. (2016). *Blockchain collaboration defines the fabric for healthcare 2.0*. Cio, Retrieved February 26, 2018 from http://cyber.usask.ca/login?url=https://search.proquest.com/docview/1780438575?accountid=14739

Noland, R. M. (2017). Intelligence testing using a tablet computer: Experiences with using Q-interactive. *Training and Education in Professional Psychology, 11*(3), 156-163. http://dx.doi.org/10.1037/tep0000149

Pearson, (2018). *Scoring.* Retrieved February 14, 2018, from https://www.pearsonclinical.ca/en/scoring.html.

Prisco, G. (2016). *The blockchain for healthcare: Gem launches Gem Health Network with Philips Blockchain Lab*, Retrieved February 14, 2018, from https://bitcoinmagazine.com/articles/the-blockchain-for-heathcare-gem-launches-gem-health-network-with-philips-blockchain-lab-1461674938/

Rabin, L. A., Spadaccini, A. T., Brodale, D. L., Grant, K. S., Elbulok-Charcape, M. M., & Barr, W. B. (2014). Utilization rates of computerized tests and test batteries among clinical neuropsychologists in the United States and Canada. *Professional Psychology: Research and Practice, 45*(5), 368-377. http://dx.doi.org/10.1037/a0037987.

Rifi, N., Rachkidi, E., Agoulmine, N., & Taher, N. C. (2017, October). Towards using blockchain technology for eHealth data access management. *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME). IEEE*.

Roalf, D. R., Ruparel, K., Gur, R. E., Bilker, W., Gerraty, R., & Elliott, M. A., et al., (2014). Neuroimaging predictors of cognitive performance across a standardized neurocognitive battery. *Neuropsychology, 28*, 161–176.

Sackett, P. R., Lievens, F., Van Iddekinge, C. H., & Kuncel, N. R. (2017). Individual differences and their measurement: A review of 100 years of research. *Journal of Applied Psychology, 102*(3), 254-273. DOI: http://dx.doi.org/10.1037/apl0000151

Sarwal, A., Insom, P. (2018). *BitHealth*. Retrieved February 26, 2018 from https://devpost.com/software/bithealth.

Shaughnessy, J. J., Zechmeister, E. B., & Zechmeister J. S. (2006). Survey Research. In J. J. Shaughnessy (Ed.), *Research Methods in Psychology (pp. 122-167)*. Beijing: Posts & Telecom Press.

Schulenberg, S. E., & Yutrzenka, B. A. (2004). Ethical issues in the use of computerized assessment. *Computers in Human Behavior, 20*, 477-490.

Schultheis, M. T., & Rizzo, A. A. (2008). *Emerging technologies in practice and research*. In J. E. Morgan & J. H. Ricker (Eds.), Textbook of clinical neuropsychology (pp. 848–865). New York, NY: Taylor and Francis.

Tama, B. A., Kweka, B. J., Park, Y., & Rhee, K. H. (2017). A critical review of blockchain and its current applications. *2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang*, 109-113. doi:10.1109/ICECOS.2017.8167115

The Apache Software Foundation. User’s Manual. Retrieved January 23, 2019 from https://jmeter.apache.org/usermanual/

The British Psychological Society. Psychological testing: A test user’s guide. Retrieved February 13, 2018 from https://ptc.bps.org.uk/sites/ptc.bps.org.uk/files/Documents/Guidelines%20and%20Information/PTC02%20Test%20Users%20Guide%202017%20WEB.pdf

The Linux Foundation. *Hyperledger composer tutorial*. Retrieved February 12, 2018 from https://hyperledger.github.io/composer/tutorials/tutorials.html.

Hyperledger Fabric. *Access Control Lists (ACL)*. Retrieved October 11, 2018 from https://hyperledger-fabric.readthedocs.io/en/release-1.2/access_control.html?highlight=permission%20control.

Tschorsch, F., & Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials, 18*(3), 2084-2123.

Vrana, S. R., & Vrana, D. T. (2017). Can a computer administer a Wechsler Intelligence Test? *Professional Psychology: Research and Practice, 48(3)*, 191-198. http://dx.doi.org/10.1037/pro0000128

Wiese, C. (2016). *Factom, Inc. Receives Grant to Create Secure Medical Records Using its Blockchain Technology*. Retrieved February 26, 2018 from https://www.factom.com/blog/gates-foundation-grant

Witt, J.-A., Alpherts, W., & Helmstaedter, C. (2013). Computerized neuropsychological testing in epilepsy: overview of available tools. *Seizure, 22*, 416–423.

Xia, Q., Sifah, E. B., Asamoah, K. O., Gao, J., Du, X., & Guizani, M. (2017). MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain. *IEEE Access, 5*, 14757-14767.

Yli-Huumo, J., Ko, D., Choi, S., Park, S., Smolander, K. (2016). Where is current research on Blockchain technology? - A systematic review. *PLoS ONE, 11*(10), 1-27. DOI: 10.1371/journal.pone.0163477

Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. *Journal of medical systems, 40*(10), 218.

Zhang, J., Xue, N., & Huang, X. (2016). A secure system for pervasive social network-based healthcare. *IEEE Access, 4*, 9239-9250.

Zhao, H., Zhang, Y., Peng, Y., & Xu, R. (2017, March). Lightweight Backup and Efficient Recovery Scheme for Health Blockchain Keys. In *Autonomous Decentralized System (ISADS), 2017 IEEE 13th International Symposium on (pp. 229-234). IEEE*.

Zuckerman, E. L. (1997). *The paperwork office: The tools to make your small psychotherapy practice work ethically, legally, profitably—forms, guidelines, and resources* (5th ed.). New York: Guilford Press.

Zyskind, G., Nathan, O., & Pentland, A. '. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. *2015 IEEE Security and Privacy Workshops, San Jose, CA*, 180-184. doi: 10.1109/SPW.2015.27