Fall 12-2020

Authentication Based on Blockchain

Norah Alilwit
Embry-Riddle Aeronautical University

Authentication Based on Blockchain

by

Norah Alilwit

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Cybersecurity Engineering
at Embry-Riddle Aeronautical University

Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, Florida
December 2020

Authentication Based on Blockchain

by Norah Alilwit

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Houbing Song, and has been approved by the members of the thesis committee. It was submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Cybersecurity Engineering.

Houbing Song

Houbing Song, Ph.D.
Committee Chair

RBabiceanu

Radu F. Babiceanu, Ph.D.
Committee Member

ILteris Demirkiran

IIteris Demirkiran, Ph.D.
Committee Member

Timothy A. Wilson
Digitally signed by Timothy A. Wilson
Date: 2020.11.19 06:21:58 -05'00'

Timothy A. Wilson, Sc.D.
Chair, Electrical Engineering and Computer Science

Digitally signed by Maj Dean Mirmirani
DN: cn=Maj Dean Mirmirani, o=Embry-Riddle Aeronautical University, ou, email=mirmiram@erau.edu, c=US
Date: 2020.11.20 16:15:49 -05'00'

Maj Mirmirani, Ph.D.
Dean, College of Engineering

Christopher Grant
Digitally signed by Christopher Grant
Date: 2020.11.23 09:42:16 -05'00'

Christopher Grant, Ph.D.
Associate Provost of Academic Support

11/25/2020
Date

Acknowledgments

I would like to express my deepest appreciation to my advisor Dr. Houbing Song for his continued help and support. His instruction helped me all the time of research and writing this thesis.

My thesis could not have been accomplished without the support of my best friend Saleh Al sidran to being beside me all the time and trying to keep me positive, encourage me to achieve my idea in really world. I am extremely grateful to you my friend.

Finally, to my loving, supportive caring dad in the world Jarallah Alilwit thank you for believing in me and encourage me to complete my study in abroad. I am extremely thankful for my mom Lolo Almubark and my sisters for their understanding and continuing support to complete this research. Also, many thanks to my brothers especially Zamil for his helpful advice.

Contents

Abstract

Across past decade online services have enabled individuals and organizations to perform different types of transactions such as banking, government transactions etc. The online services have also enabled more developments of applications, at cheap cost with elastic and scalable, fault tolerant system. These online services are offered by services providers which are use authentication, authorization and accounting framework based on client-server model. Though this model has been used over decades, study shows it is vulnerable to different hacks and it is also inconvenient to use for the end users. In addition, the services provider has total control over user data which they can monitor, trace, leak and even modify at their will. Thus, the user data ownership, digital identity and use of online services has raised privacy and security concern for the users.

In this thesis, Blockchain and the e-pass application are studied and alternative model for authentication, authorization and accounting is proposed based on Ethereum Blockchain. Furthermore, a prototype is developed which enables users to consume online services by authenticating, authorizing, and accounting with a single identity without sharing any private user data with the services provider center server. Experiments are run with the prototype to verify that it works as expected. Measurements are done to assess the feasibility and scalability of the solution. In the final part of the thesis, pros and cons of the proposed solution are discussed and perspectives for further research are sketched.

1.0 CHAPTER 1: INTRODUCTION.

Nowadays, most people use the smart services that are provided by services providers, such us online banking which allows the users to do their daily transactions. This facilitated access to services directly without the need to visit the services provider physical location (Trappe, 2018). As we have seen recently, most of the world's governments are moving towards being smart and flexible governments that serve their citizens through smart channels, for example, now the citizen can apply to open his\her own commercial file without needs to visit the country's economic institution and can apply for a passport without visit the Immigration and Passports headquarters. All these transactions require the user's authentication and identity verification to avoid any personal fraud and keep the user’s rights and according to the survey, it was found that most of the providers of this type of services depend entirely on the traditional authentication methods , such as the user name, password, personal identification code, PIN, OTP and security questions based on the user's previous answers which is increasing the users' fears of sharing their personal and financial information, especially since recently there has been a lot of news about hacking accounts and leaking data to millions of active accounts of famous websites such as Alibaba and Twitter (Nia, 2017). The most important issue that we solved in this thesis is the sensitive transactions on behalf of the real user and the self-contained user data ownership and use several services provider services as well as pay invoices without sharing the user private data. This is important issue because this type of attack is very difficult to tracing and control. The attacker has several method and tools to obtain the victims personal data and sensitive details and the most famous tool is social engineering method (Korshid, 2016).

Today, the most dangerous issue in the internet when the identity theft because when the theater already has the victim credential that presenting the user identity, the attacker can make any sensitive transactions on behalf of the real user. This type of attack is very difficult to tracing and control. The attacker has several method and tools to obtain the victims personal data and sensitive details and the most famous tool is social engineering method. The target system not smart enough to deleted whatever this transaction is done by the right user or someone on behalf, hence this make the online transactions not secure and untrusted (Patil, 2018). This thesis addresses how the public users can have self-contained and secure their information ownership on several services providers servers as well as have secure channel for their authentication between the user and the provider without needs to share the users sensitive data to every service provider (Madhusudhana, 2019). Another common problem is the inaccuracy of the data for example, the user is the real user and has right to access the services by applying on it but some by mistake or intentionally, wrong information is provided, such as the tax file number or the user's national number, which causes inaccuracies and identical data in systems that do not have a strong and flexible infrastructure that handles all these accidents and scenarios. As we know today, there are many services that are available through the Internet, and most of the service providers, whether governmental or private, are looking to provide smart services to their customers to be distinguished by their services and to attract customers and investors alike. If one person needs to present a travel approval, for example, or renew a national identity, he needs to create an account for each of the service on the service provider platform. Every service provider needs user information and the login details of a person, so he cannot remember all these accounts if he has twenty accounts, for example, and he may face real difficulty in remembering. He is forced to reset his password every time he performs on a specific service, which is cumbersome and annoys the user (Sandhu, 2019).

In the thesis, I had triggered a big debate on user data privacy and about security of the user's digital identity. My proposal an (E-pass) platform using blockchain technology to address the online user's privacy and identity. The prototype authenticates, authorize and accounting without requiring users to share their private data to the providers center server. By using blockchain, will verifies and ensures that the users, transactions, messages are legitimate. Authorization is done by smart contracts which are written and deployed to blockchain. At the same time, it provides a common distributed, decentralized identity backend where services providers can deploy their own authorization logic and validate the user identities. This solution is suitable and logical for user authentication that allows users to login, register, and pay their invoices online. Moreover, this solution is providing scalability, elasticity, reliability, and redundancy as well. There has been a huge push around the unified digital identity and technologies that support security that have garnered the interest of researchers and governments in recent years. Decentralization is expected to become a necessity and imperative in all digital solutions provided by governments and private enterprises that request private and sensitive information from their clients.

The importance of the thesis over its predecessors is that it used an existing product and an existing technology to develop a new solution. From the results of the research it was found that some countries in the region, such as the United Arab Emirates, have already used the digital identity of citizens recognized as a legal user identity on government services platforms, but this identity lacked the use of modern technologies to support this identity and make it more secure and reliable, because only the central server performs the authorization and authentication, on the contrary, here all the network points must approve the authorization transaction by applying the blockchain technology that takes advantage of decentralization (M Smits, 2020).

In this thesis, the motivation, problem statement with research methodology and related work are discussed. Below chapters will describes the key concepts of authentication, authorization, accounting along with possible vulnerabilities. Furthermore, covers the key concept of blockchain technology, its architecture, how it works and its vulnerabilities and describes the most popular blockchain applications. Also, will describes the reason for selecting an Ethereum blockchain for implementing and describes the prototype design with software architecture and flow diagram and describes a prototype implementation with hardware and software components along with its environment setup and actual implementation and execution. And last section covers the testing of the prototype and finally, the discussion is presented which is followed by the conclusion and discussion of possible future work.

2.0 CHAPTER 2: LITERATURE REVIEW.

Service provider is the entity which offers online services to public such as commercial licenses, official documents, banking transactions. that can be rapidly provisioned with minimal effort (Dean, 2014). Most of these providers using traditional authentication login methods to authenticating users which is based on authentication, authorization. This is using for controlling the access of services, enforcing policies, auditing by compares a user's authentication credentials details that provided by users in login process with the user credentials information that been stored in a provider database. User is have right to access the service once the provided credentials match the providers database records. This traditional method of authentication carries a lot of security problems, such as leakage of user information and fraud on the system by using another users' information. Authorization is process of determining whether the user has authority to access the requested content or services. It is tightly coupled with the authentication as the user must be authenticated to get authorized. Authorization is required because in the online service environment, same services might be consumed by different users who have different access control rights. For example, access rights for citizen and resident is different. Thus, online services environment must have police to allow access to services which belong to the user (Vishnia, 2020).

As we mentioned in the introduction about the problems that may facing the traditional authentication method, in this proposal we propose a smart and secure solution for authentication that protects user rights, prevent tampering with their personal information, and system fraud. This proposal based on E-pass platform that provides secure solitons for both users and services provider. Users can register themself on the E-pass platform with national-Id that makes the user authenticated for all services provides such government smart serves and the private entities as well. For all entities that want integrated the E-pass solution with their login option, they should register them self on the E-pass platform as well as users to able to get integrations parameters which are entity short name and entity Id by make agreement between the E-pass platform and the entity. Once user successfully registered on the E-pass platform and want to do login with the entity that already made agreement and integrated with the E-pass plat form, they can click on the login with E-pass icon then the webpage will ask user to enter the national Id or registered mobile number, once the platform found the registered user will sent OTP as secure PIN to registered mobile number and ask the user to enter the received OTP to complete the authentication process , the platform will return to the service provider the user is authenticated and requested details. This E-pass platform uses the blockchain technology to make this authentication transaction as describing in the below workflow figure.

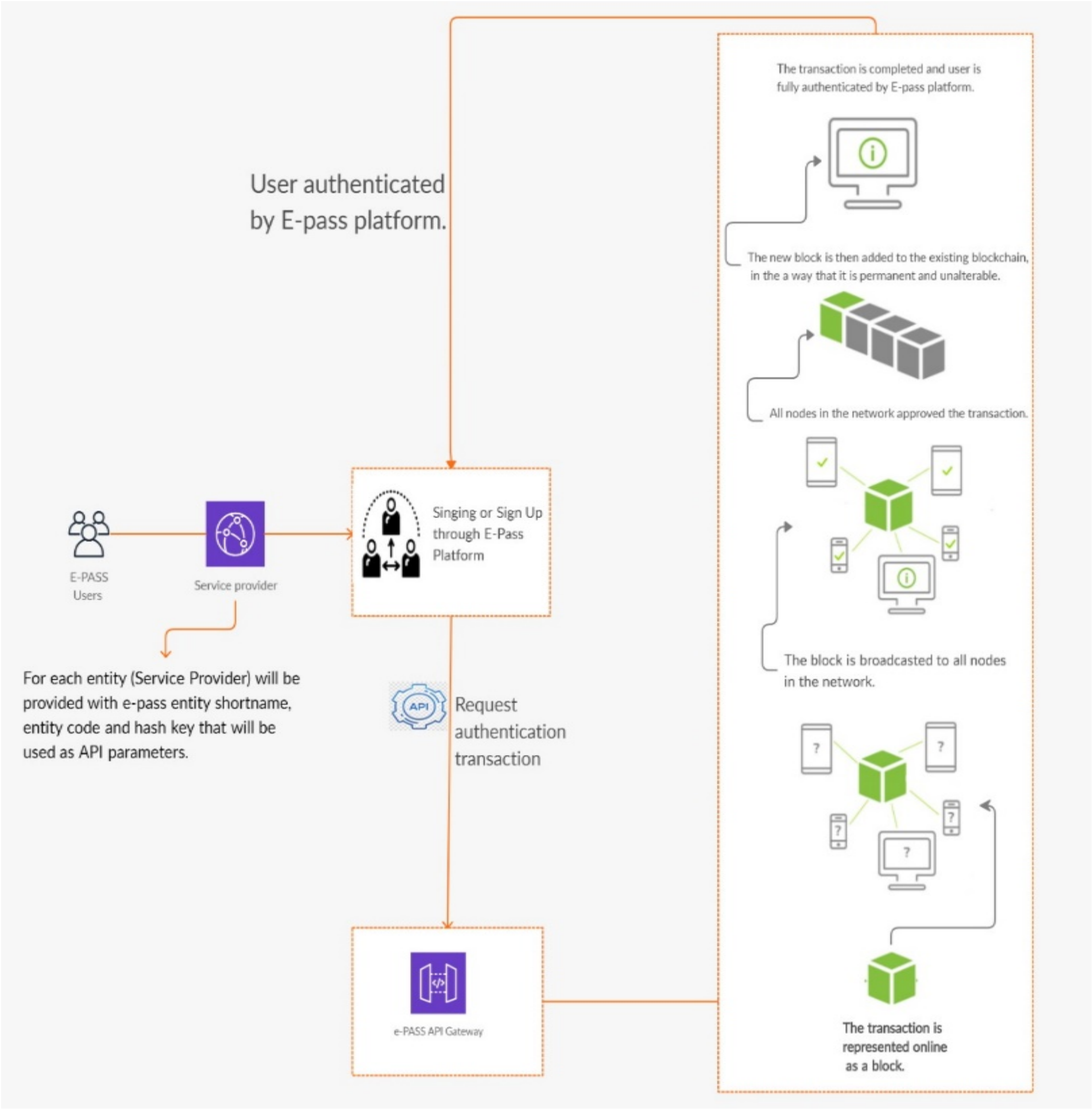

User authenticated by E-pass platform.
The transaction is completed and user is fully authenticated by E-pass platform.
The new block is then added to the existing blockchain, in the a way that it is permanent and unalterable.
All nodes in the network approved the transaction.
E-PASS Users
Service provider
Singing or Sign Up through E-Pass Platform
For each entity (Service Provider) will be provided with e-pass entity shortname, entity code and hash key that will be used as API parameters.
API
Request authentication transaction
The block is broadcasted to all nodes in the network.
?
e-PASS API Gateway
The transaction is represented online as a block.

Figure 1. E-pass over all flow work.

This solution will help users to access all governments and privates' online services through one platform, in this way users no need to share sensitive personal detail to many platforms and there will be no need to multiple usernames and passwords (Polese, 2019). E-pass will also allow digital signatures to legally complete transactions or obtain ownership of assets, so there will be no need for service center visits to process any paperwork.

Also, in future can use this platform to start a business, buy a staff or rent a car in a few clicks. This platform will allow residents to register for the first national digital smart identity, using a password-less and paperless secure profile to access government and commercial services, and sign, verify and share digital versions of documents. This solution will be middlemen-free attack due the decentralization that no need of a middleman and also this solution come with censorship-free due that the blockchain network is not controlled by a single party but with every network node's participation. This proposal provides a smart authentication solution for a client and services providers with full benefits of new blockchain technology. This will enable secure access and fast login for clines authorization and authentication. Also, this will help on payments with more secure details, users no need to gives the bank card details because the solution is based on distributes self-sovereign system with no single point of failure and all about decentralized.

Services provider is the entity which offers online services to public such as commercial licenses, official documents, banking transactions, etc. that can be rapidly provisioned with minimal effort. Authentication, Authorization and Accounting (AAA) is a framework used by the services provider for controlling the access of services, enforcing policies, auditing. The services provider must use this framework for effective resource, user, network, and security management. The AAA is based on client-server model where the user interacts with client and server has the business logic necessary for services resources, user, network, and security management. Authentication is a process of verifying the identity of the user and authorization is the process of deciding whether a user has enough rights to use the requested service. Accounting is the process of tracking the services applied by the user for billing, auditing, data analytics.

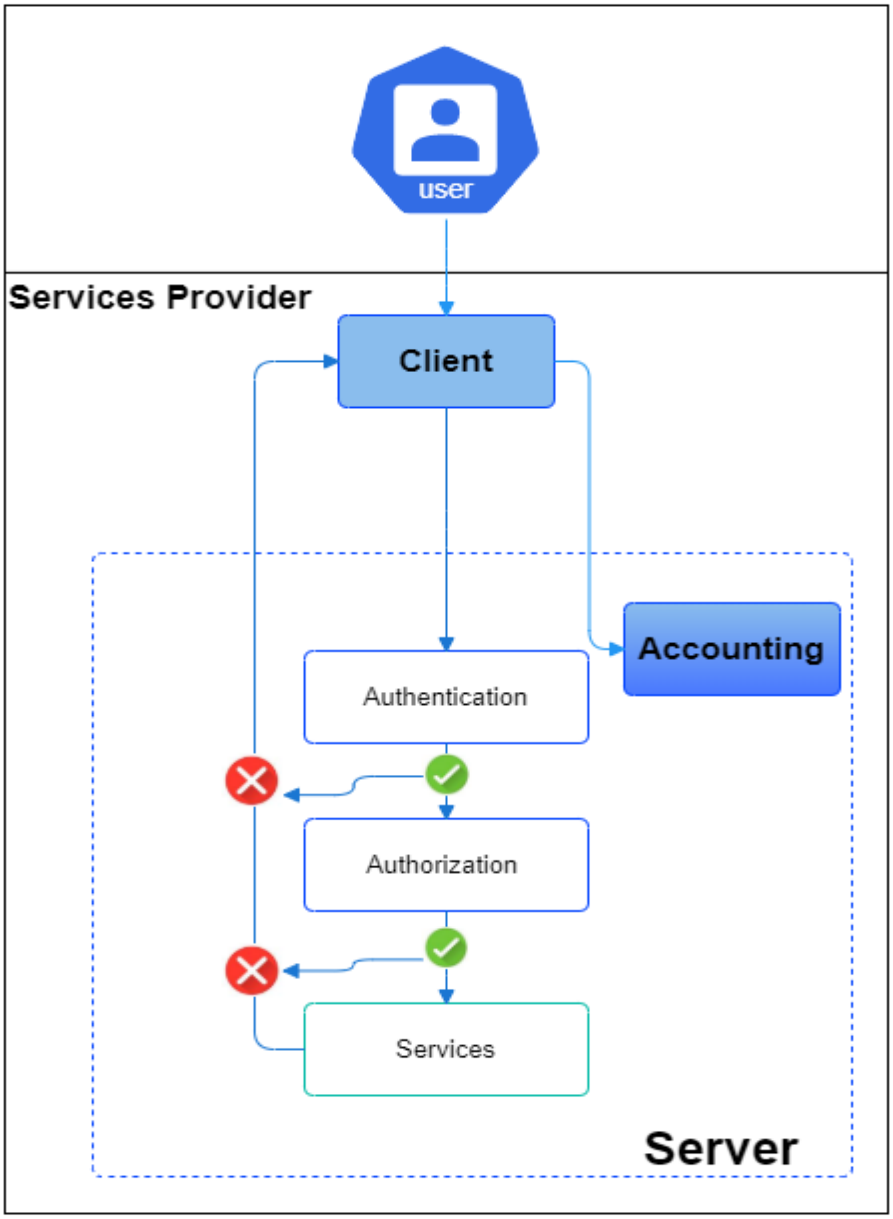


Figure 2. AAA Platform overview.

Above diagram shows the general architecture with client-server model where client is a web or mobile application and server host authentication, authorization, accounting services, which is divided into two parts where the first part has a services user and the second part has a services provider. The provider requires authentication and authorization from the user. The user interacts with the provider with client application which sends the request to the server. The server has authentication and authorization services. The authentication service verifies the credential of the user. If verification is successful, the request is forwarded to the authorization service. Otherwise, an error is returned, and the user is redirected to the client application. The authorization service determines the authority of the user. If successful, the request is forwarded to the service that returns the requested service by the user. Otherwise, error is returned, and user is redirected to the client application. The accounting service intercepts the request between client and server and does the metrics calculation and audition as provisioned by the provider.

2.1 Authentication.

Authentication is a mechanism by which a provider identifies the user before granting access to the services. The provider enables the user to use the services based on the credentials provided during the registration. The provider authentication can be categorized into three categories: what-you-know (knowledge), what-you have (possession) and who-you-are (ownership). What-you-know means something what the user knows about such as username and password, PIN code and public keys. What-you-have is something what user possess such as smart cards, identity card, e-tokens (identity information encrypted on a flash card). Who-you-are means something that the user owns such as biometric characteristics such as fingerprints and iris scan (Kang, 2017).

2.2 Authorization.

Authorization is process of determining whether the user has authority to access the requested content or services. It is tightly coupled with the authentication as the user must be authenticated to get authorized. Authorization is required because in the online services environment, same services might be consumed by different users who have different access control rights. For example, access rights for citizen and resident is different. Thus, online services environment must have policy to allow access to services which belong to the user. The most popular authorization system is eXtensible Access Control Markup Language (XACML), and the most popular authorization framework is OAuth 2. Which is authorization framework that allows a third-party application to obtain limited access to a resource on behalf of resource owner and with owner consent (Vishnia, 2020).

The access is requested by the client or third party-application which can be a web service or a mobile application. For example, a web service might use Facebook login which basically allows the user (resource owner) to use Facebook credential to authenticate and authorize the user for accessing the web service resources. This framework is successor of OAuth 1.0 and is used by companies like Facebook, Google and Microsoft OAuth 2.0 defines four roles: resource owner, resource server, client, and authorization server. Resource owner is an entity/person who grants access to a protected resource and resource server is a server which hosts the protected resource. The client is an application which accesses the resource on behalf of the user. The authorization server is the server which issues the access token to the client after successful authentication of the resource owner. This token acts as user identity and is valid for certain interval of time, like 24 hours. The authorization flow starts with trigger from resource owner who wants to access the protected resources. The trigger is followed by client (application) which asks authorization from the resource owner and returns the authorization grant on success. On success, the authorization server returns the access token, and this access token is then sent to the resource server which validates the token and on success returns the protected resource. Thus, a resource owner (user) can access the protected resource without providing credential to the client.

2.3 Accounting.

This is the process that keeps track of a user's activity while attached to a system; the trail included the amount of the outstanding, the services accessed, and how much data transferred. Accounting services applied is used for trending, detecting breaches, and forensic investigating.

Keeping track of users and their activities serves many purposes. For example, tracing back to events leading up to a cybersecurity incident can prove very valuable to a forensics analysis and investigation case. Identity Management Systems such as Aruba's ClearPass and Cisco's Identity Services Engine (ISE) utilize the AAA framework via RADIUS, TACACS, and other mechanisms. Identity Management and Network Access Control are two important tenants of a sound security policy (Zwitter, 2020).

2.4 Potential Vulnerabilities.

Although AAA framework has been used over a decade by services providers, it has various major potential vulnerabilities such as distributed denial of service (DDoS) attacks, brute force, man-in-the-middle (MITM), account hijacking and data breach. According to 2017 McAfee (a computer security software company) report of Net Losses and the global cost of cybercrime, there would be the global economic loss of between $375 to $575 billion each year which is more than the national incomes of most countries and governments (Dean, 2014).

2.4.1 Account Hijacking.

Account Hijacking is an attack where a malicious user (attacker) can retrieve the user credential and use it on attacker’s favor. With this attack, an attacker can eavesdrop on the user transactions and activity, modify their information, redirect to unauthorized sites, and return fabricated data (Treiblmaier, 2019).

2.4.2 Distributed Denial of Service attack.

Distributed Denial of Service is an attack where a group of compromised malicious users flood a services provider with unnecessary request traffic. The attack results in the denial of service for the legitimate user. The unnecessary traffic might slow down or even crash and shut down the provider services (Polese, 2019).

2.4.3 Man-in-the-Middle attack.

Man-in-the-Middle attack is an attack where a malicious user (attacker) can intercept and modify the communication between two systems. An attacker listens to the traffic between user and service and eventually splits the original traffic into 2 new connections where one connection is between user and attacker and another between the attacker and the service. Thus, the attacker acts as a proxy and can read, insert, and modify the intercepted data (Trappe, 2018).

2.4.4 Data Breach.

Data breach has been one of the primary causes of online cyber-theft, account hijacking and fraud in services computing. The primary reason for data breach is the centralized servers of the services providers which creates a honeypot for hackers where they can execute multiple vulnerabilities such as phishing, denial-of-service, backdoors, spoofing, clickjacking, MITM (Cheng, 2017).

2.4.5 Malicious Insiders.

Malicious Insiders are threats which occur because of lack of transparency in the service provider process and policy compliance. A service provider might monitor their employees or users without their consent and grant access to their data to third party organization (Akanbi, 2020).

2.5 Drawbacks.

The major drawbacks of the current AAA framework are user data ownership and high cost to deploy and operate AAA systems. The main reasons behind these drawbacks are the AAA's client-server architecture adopted by the service providers and service level agreement (SLA). SLA is an agreement between the service provider and the user which defines the terms and conditions for the provisioning and delivery of the services including security measures. Also, with SLA, the right to use user data is granted to the service providers. Hence, with client-server architecture the user data gets saved to the central servers and with SLA the service providers have full access to these data which they can monitor, trace, leak, and control. This results to the potential user data vulnerabilities and fraud as discussed in above. Thus, though the data belongs to the user, it is eventually controlled by the providers which is one of crucial issues of data privacy and security. Moreover, the SLA requires the provider to guarantee no downtime in the services. So, to achieve this, the provider must maintain data centers which would ensure almost 100% uptime for the services. The maintenance and operation of these data centers is expensive as it requires lots of hardware for computing, network, storage, firewall as well as cooling system with operations to ensure 24/7 support of the system (Vishnia, 2020).

2.6 Blockchain technology.

A blockchain is a common database, ledger or database of many transactions that are saved on multiple points in different locations. The database grows constantly when new transactions or "blocks" are added to it. This forms a continuous chain of data where records are public and verifiable. Since there is no central location, it is difficult to penetrate the information found in the millions of different locations of the blockchain. Blockchain technology is an entirely new software technology that first appeared in 2009 in a research paper presented by Satoshi Nakamoto, whose purpose was primarily to create the digital currency Bitcoin (Treiblmaier, 2019).

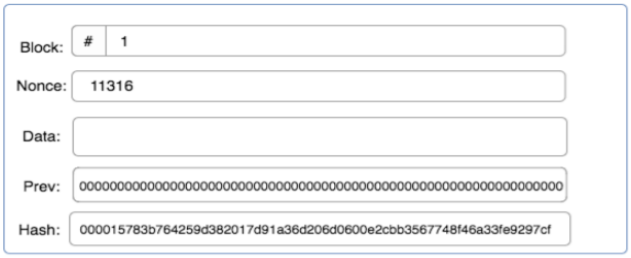
Blockchain is not adjustable, as any change to it requires immense computing power, and the blockchain becomes more secure the older it is. Blockchain is characterized to some extent by transparency, as anyone can view the data stored in the blockchain (Bitcoin for example) that all stored transactions can be viewed using the blockchain browser. However, some blockchain technologies provide more anonymity. Blockchain tends to be decentralized, as there is no central authority controlling it, unlike traditional databases that can be blocked and monitored by its owner. Blockchain can maintain its effectiveness 24/7 in the event of any disruption to the network. However, there are trends of more centralized blockchain projects.

Moreover, there are two types of blockchain which are public blockchain like the Bitcoin that anyone can join the contract network and the private blockchain which is often used by companies where only people who have permission can add their own devices to the network as blockchain nodes. In this thesis, we use blockchain technology to make the authentication transaction between users and services providers for secure access, user authentication, and user identification of the service provider's system. The thesis is presented a single trusted digital identity for all citizens that aiming to provide a single trusted digital identity solution for service providers, while maintaining a high level of security assurance and seamless user experience protected using the blockchain technology (Charles, 2019).

There are five terminologies that been used in new blockchain technology as following:

A peer-to-peer network is the process of exchanging files and data between two points such computers on the network. The peers provide a portion of their resources, such as processor capacity, disk storage space, or network traffic (data transfer speed), and is directly available to other network participants, without need for central coordination by servers or static hosts. And we use the blockchain technology in this proposal due that the blockchain uses peer-2-peer network architecture to ensure the network is distributed, decentralized with no point of failure, hence this advantage will make the solution more availability (Riva, 2020). Second terminology is a block which is a one and only one unit in the blockchain technology and is composed of transactions with meta-data.

As we can see in the below sample block Figure where there is a block number for each black, data, hash and nonce. There is a primary key of the block which is the block number #1 that will be unique of each block, other is the nonce which is random arbitrary number guessed to find the specific format of the hash. Last, the Data is the user digital data which is empty in this case and the Hash is the digital fingerprint of the data.



Block: # 1

Nonce: 11316

Data:

Prev: 0000000000000000000000000000000000000000000000000000000000000000

Hash: 000015783b764259d382017d91a36d206d0600e2cbb3567748f46a33fe9297cf

Figure 3. Block structure.

Distributed Blockchain is blockchain distributed over P2P blockchain nodes where all the nodes have an exact copy of blockchain. Thus, if one entry in blockchain is modified and re-mined, the resulting hash becomes different compared to other nodes. As a result, this transaction will get invalidated since other nodes will invalidate this copy. However, a miner could theoretically modify one blockchain entry and re-mine all hashes entry over distributed nodes if it has more computational power than other miners combined (Zwitter, 2020). Smart contracts are self-executing autonomous computer programs that get executed based on condition defined by programmer. These contracts can facilitate, enforcing and executing agreements between two parties using blockchain. Unlike traditional contracts, where a third party (bank, notary) is required, smart contracts enable independent business between anonymous parties with cheaper fees. For example, one can pay room rent automatically at the end of month without involving a bank in between. Smart contracts have various possible applications such as trading or loaning of properties, stock, or bond trading in distributed markets. Moreover, it can be also used for autonomous transparent digital voting system or autonomous digital notary contract system. In this context, companies like Ethereum, Codius are enabling smart contracts using blockchain to support these applications (Zwitter, 2020).

2.6.1 Terminology.

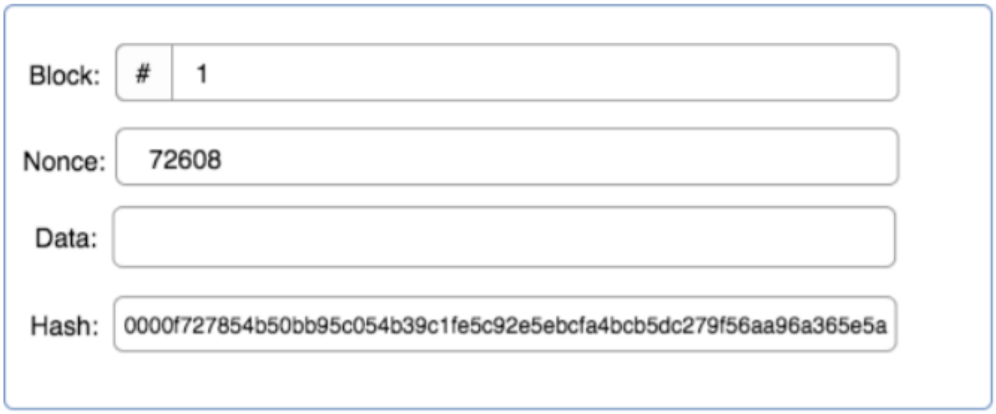
These terminologies are Peer-to-Peer (P2P) network, block, blockchain, distributed blockchain and smart contracts.

2.6.1.1 Peer-to-Peer Network.

Peer-to-peer (P2P) is distributed network architecture where each participating node (computer) share its hardware resource such as computing, storage capacity, network links with each other. Moreover, these resources are used to provide services like content and file sharing and are available to all nodes directly without need of any central server. Besides this, at given time, a node can be consumer as well producer of the resources (Sandner, 2019).

2.6.1.2 Block

A block is a single unit in the blockchain which is building block of blockchain and is composed of transactions with meta-data. A miner collects the valid data (transactions) of certain time interval to form a block and calculate the cryptographic hash. However, this hash must be specific format such as the hash must have leading four zeros. To get this specific type of hash, the miner must randomly guess an arbitrary number which outputs the hash with four leading zeros. This arbitrary number is called number used once or number once (nonce) and the block with nonce is called signed block else it is unsigned. Also, the process of finding nonce is called mining (Shackelford, 2016).



Block: # 1

Nonce: 72608

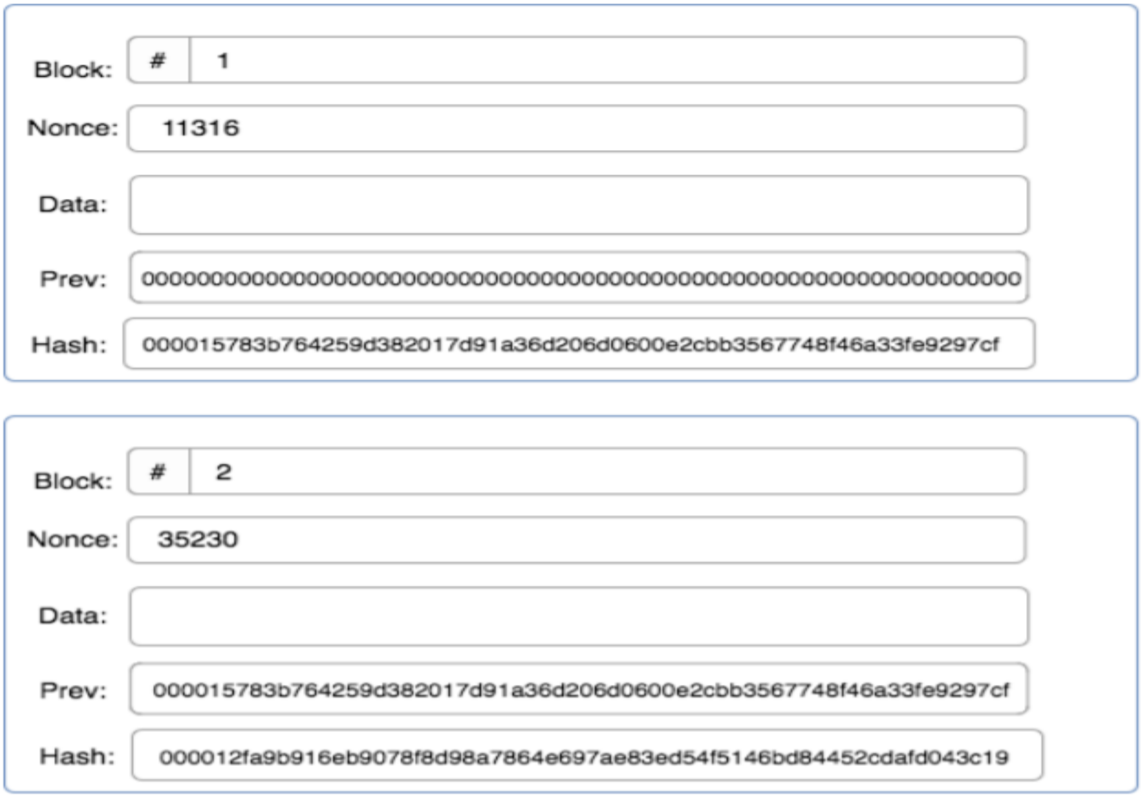
Data:

Hash: 0000f727854b50bb95c054b39c1fe5c92e5ebcfa4bcb5dc279f56aa96a365e5a

Figure 4. Block structure #1.

2.6.1.3 Blockchain.

Blockchain is a data structure with linked lists of hash pointers. It is a chain of blocks where each block has a hash pointer to the previous block. This hash pointer allows to verify and validate the digest of previous data. If any value in the chain is changed the digest of that block and the hash pointer of following blocks will change. Thus, this creates tamper-evident log which cannot be changed. Moreover, the hash pointer can be followed till the very first block called the genesis block of blockchain (Trappe, 2018).



Block: # 1
Nonce: 11316
Data:
Prev: 0000000000000000000000000000000000000000000000000000000000000000
Hash: 000015783b764259d382017d91a36d206d0600e2cbb3567748f46a33fe9297cf

Block: # 2
Nonce: 35230
Data:
Prev: 000015783b764259d382017d91a36d206d0600e2cbb3567748f46a33fe9297cf
Hash: 000012fa9b916eb9078f8d98a7864e697ae83ed54f5146bd84452cdafd043c19

Figure 5.Two blocks are chained together to form a blockchain.

A sample blockchain is shown above where two blocks are chained together to form a blockchain. Here, each block has a block number, nonce, data, previous block hash and current block hash. Moreover, Block #1 is the genesis block of the blockchain, and the hash of previous hash pointer is null (0000000000000000000000000000000000000000000000000000000000000000) integer while Block #2 as shown above is the second block of the blockchain which has the hash pointer of the previous block.

2.6.1.4 Distributed Blockchain.

Distributed Blockchain is blockchain distributed over P2P blockchain nodes where all the nodes have an exact copy of blockchain. Thus, if one entry in blockchain is modified and remined, the resulting hash becomes different compared to other nodes. As a result, this transaction will get invalidated since other nodes will invalidate this copy. However, a miner could theoretically modify one blockchain entry and remine all hashes entry over distributed nodes if it has more computational power than other miners combined (Kang, 2017).

2.6.1.5 Smart Contracts.

Smart contracts are self-executing autonomous computer programs that get executed based on condition defined by programmer. These contracts can facilitate, enforcing and executing agreements between two parties using blockchain. Unlike traditional contracts, where a third party (bank, notary) is required, smart contracts enable independent business between anonymous parties with cheaper fees (Hoffmann, 2019) .

2.6.2 Cryptography

This subsection describes about the basic cryptography technologies used in the blockchain. They are cryptographic hash function, hash pointer, digital signature and Merkle tree

2.6.2.1 Cryptographic Hash Function.

Cryptographic hash function is a mathematical function that takes any input string (data) of any length and outputs fixed sized alphanumeric string. The output string is called hash value or digest or digital fingerprint or checksum. Moreover, the output is of fixed length and unique. The function always produces the same hash from the same data despite the number of times recalculated. The hash cannot be reversed to get the input data and therefore, it can be used to check the integrity of data (Gauravaram, 2018). Thus, it is also known as one-way hash function. The hash function has three main properties, namely: collision free, hiding and puzzle friendly. Collision free means it is extremely unlikely to find two different messages that have the same hash. For example, the hash of a string x and the hash of string y are always different, despite how many times it is calculated. Hiding means it is infeasible to find x from given hash of x and puzzle friendly means it is easy enough to calculate a hash of given data (Treiblmaier, 2019).

2.6.2.2 Hash Pointer.

Hash pointer is a pointer to where data is stored along with digest of that data. In other words, it is just a hash that is used to reference another piece of known information which can be used to verify the data digest (data has changed or not). The hash pointer can be used to build data structures like blockchain which is a linked list of hash pointers and Merkle tree which is a binary tree of hash pointers (CHEN, 2015).

2.6.2.3 Digital signature.

Digital signature is another building element of the blockchain. It uses public-key cryptography to provide the integrity, nonrepudiation (obligation of message sent and received by the parties) and authenticity of a message and its source. It has similar properties as a manual signature which can be issued only by the issuer and which is verifiable by other users. A message signed with a digital signature can be verified by other users, but the message can be signed only by signature owner. Beside this, digital signatures are created using public key cryptography. Public key cryptography or asymmetric cryptography uses a key which is a combination of public and private key. The private key is saved only by the owner while the public key is distributed to the other user (Mets, 2017).

2.6.2.4 Merkle Tree.

Blockchain uses P2P network where each peer must have same copy of data and new data must be propagated and verified across the network. Propagating and verifying data over P2P network is time consuming and computationally expensive. Therefore, Merkle tree is used which instead of sending data only the hash of the data is sent, and the receiver peer checks the hash against the root of the Merkle tree which allows secure and efficient verification of larger data structures as well as ensures data integrity. Merkle tree or hash tree is binary tree of hash pointers which ensures all the peers/nodes must have same undamaged, unaltered, legitimate data and if a data is changed in one node, changes must be propagated to every node. Merkle tree is composed of many blocks containing data or transactions. These blocks from the leaves of the Merkle tree, and the transaction blocks are grouped into pairs of two where each pair has hash pointers respectively which eventually, make the next level up of the tree. Moreover, this process is repeated until the single block is reached and the single block is called root hash or the root of the tree (Jal, 2018).

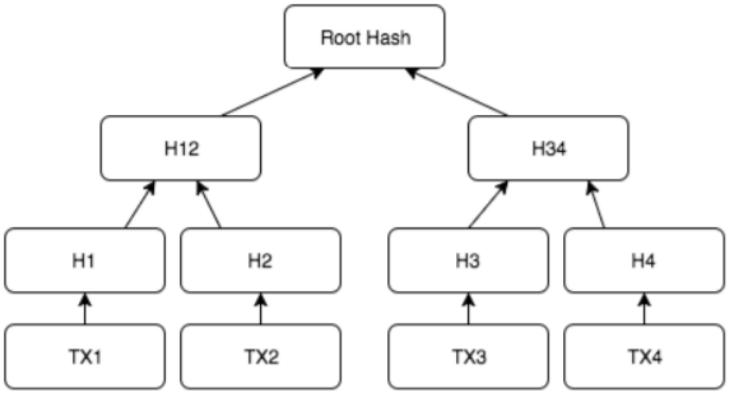


Figure 6.Merkle Tree.

It has four transactions TX1, TX2, TX3 and TX4 at the bottom where each of the four-transaction data pass through a hash function to generate four unique hashes. Moreover, pairs of hashes are then combined and passed though hash function which generates two unique hashes and the two hashes are then combined and again passed to hash function which generates one unique hash which results in the root hash forming complete Merkle tree.

2.6.3 Architecture.

The architecture is divided into four layers: Application, RPC API, Peer Service and Consensus Layer. Application Layer has the blockchain applications which enable users to interact with blockchain network. The blockchain applications are blockchain wallet or blockchain key management software such as ledger wallet which provides blockchain identity to the user by creating public/private key-pair where public key is shared to the network and private key remains only with the user. Moreover, a transaction is created in this layer. The transaction contains sender’s digital signature, value of the transaction and receiver’s public key. This transaction is signed bypass-phrase protected sender’s private key which creates the unique identifier of the transaction called transaction hash. Depending on the security measure taken by the sender it can also be biometrics (fingerprint, iris) or password protected (Goodell, 2018).

Thus, this layer on one hand ensure data integrity of transaction and on other hand provides anonymity to the users. Moreover, the user private keys remain only with the user and no personal data (first name, last name, email, phone number) is shared to the network. Other blockchain applications are web applications such as BitInfoCharts and Blockchain info which provides cryptocurrency statistics as well as interface where any user can query and view transactions. Beside these, the user can develop their own blockchain applications using RPC API Layer. RPC API Layer is a communication layer between Application Layer and Peer Service layer (McConnell, 2016).

These rules depend on the blockchain environment setup, either development or testing or production. Blockchain specifically uses JSON-RPC API, which is simple, transport agnostic, stateless, light-weight remote procedure call. Thus, the Application Layer can access only those APIs which are allowed by the RPC API

Layer. Furthermore, there are various peers requesting and sending data from blockchain network. There are special nodes called miners that form the Peer Service Layer. The Peer Service Layer is the core of the blockchain which performs all the computing necessary for validation and verification of transactions and blocks. Each node has copy of the transaction history which can be verified till the first transaction. This makes the blockchain decentralized, distributed, fault tolerant and resilient since there is no single point of failure and data is distributed over multiple nodes. It has the communication and access rules of the blockchain which are pre-defined by the blockchain network. These rules define the endpoint of the database, personal, network, admin, personal, metrics and debug API interfaces of the blockchain. It also sets the rules about which API are exposed to the Application Layer from Peer Service Layer as well as who can access the application (Banach, 2020).

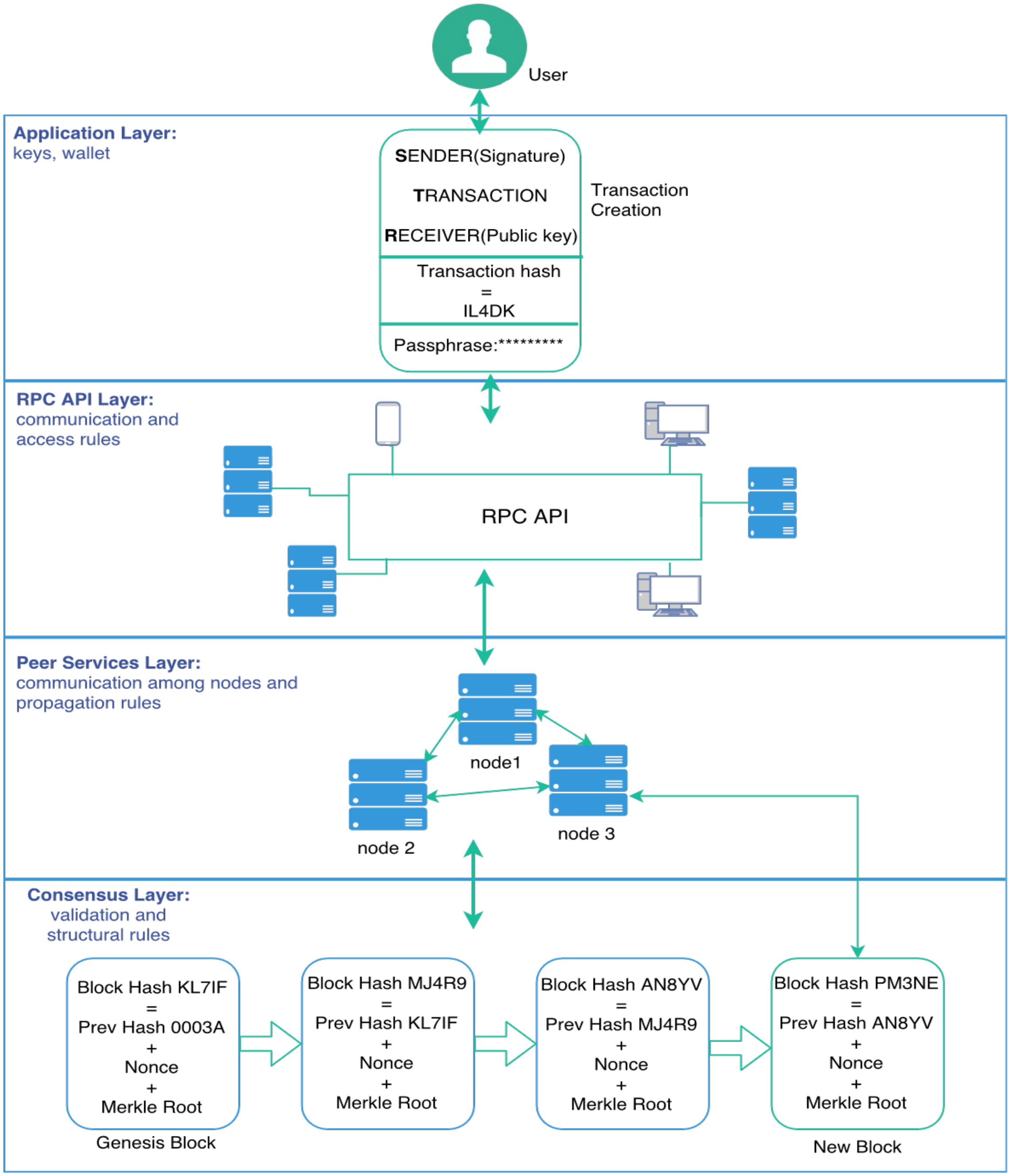


Figure 7.Block chain Architecture

Moreover, a node in this layer validates if the new transaction received from the Application Layer is of right format, is signed by sender digital signature, is feasible (has enough balance to perform transaction). The node also authenticates the sender by checking his digital signature. Finally, the node creates a new block based on the validation and structural rules defined on the Consensus Layer. The Consensus Layer validates the structure of block created by the node. The block must have a valid block hash, previous block hash, nonce and Merkle root. The block hash is the unique identifier of the new block, previous block hash is the hash pointer to the previous block, nonce relates to proof-of-work and it is result of the solution of mathematical puzzle set by consensus layer. Finally, Merkle root enables verification of all the transaction till first transaction. The node gets financial reward for the proof-of-work and the new block is immediately broadcasted to the network where other nodes updates their own ledger resulting to consensus on the network. Thus, the consensus layer makes the blockchain transparent, immutable audit trail. If one transaction is changed on one node, all other nodes must update this change and re-mine the blocks. Hence, the digital ledger cannot be manipulated, and fraud is very difficult (Madhusudhana, 2019).

2.6.4 How blockchain works.

Blockchain workflow has five steps which are Transaction Definition, Transaction Authentication, Block Creation, Block Validation and Block Chaining. Transaction Definition is the model of the transaction pre-defined by the blockchain network. It has sender digital signature, the transaction payload and receiver’s public key which is cryptographically signed with sender’s passphrase protected digital key. Transaction Authentication is the process by which the nodes validate if the A has the asset, enough balance to send the asset and is authenticated to move the asset. Block Creation is process of creating block by a node from the transaction pool where transaction is grouped together based on the creation time. Block validation is process of validating the block by checking if it has previous hash and nonce which provides the proof-of-work. Block Chaining is process of adding the block to the blockchain once the nodes reach on the consensus (Pech, 2017).

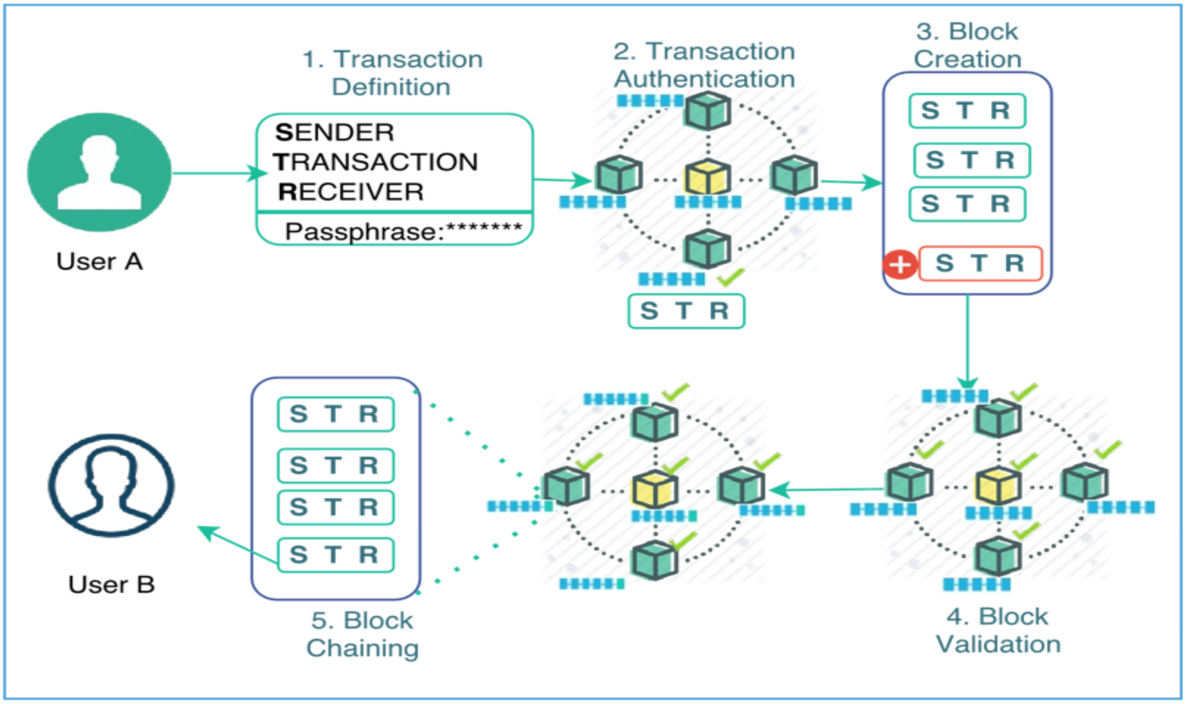


Figure 8. Blockchain workflow

2.6.5 Potential Vulnerabilities.

Blockchain is faster and cheaper than centralized system because of its decentralized distributed design. Although, it is reliable and secure because of its consensus protocol, cryptography, and anonymity, it still has several potential vulnerabilities such as the 51% attack, sybil attack, identity theft and code-based attack (Hawkins, 2014).

2.6.5.1 The 51% attack.

A 51% attack is when a miner or a mining pool controls 51% of blockchain network computation resources. As a result, they can dominate the validation and verification of transactions as well as they can change the content of blockchain. Moreover, they could invalidate the valid blocks, create, and confirm their own fraudulent blocks eventually quicker than the rest of honest miners which could result to double spending. Besides this, the attackers can change the consensus rule, steal assets from others and even prevent cryptocurrency generation. So far, no bad incident has happened with 51% attack although in July 2014, mining pool ghash.io held more than 51% of bitcoin blockchain network. But, shortly after other miners moved out of this mining pool, hence any tragedy was avoided. This attack can be prevented using proof-of-stake consensus algorithm (Vishnia, 2020).

2.6.5.2 Sybil attack.

Blockchain does not have a central authority to administer identities of the participants. As a result, attacker can create multiple copies of itself, which might look like separate participants though they are all controlled by the same node. The attacker can try to fill the network with its clients. So, other nodes are likely to connect only to attacker nodes. The attacker can then refuse to relay blocks and transactions from others, disconnect the connecting node from the network or relay only blocks created by itself. This attack can be avoided by only trusting the blockchain with the most proof-of-work since it cannot be easily faked because of the significant mining power requirement (Ha, 2017).

2.6.5.3 Identity theft.

Although blockchain provides the ownership of the user identity, this digital identity is backed up by the private key that must be kept safely. If the private key is stolen or device storing the private key gets hacked, the victim will lose all its digital assets as well as its digital identity. Moreover, this digital identity cannot be recovered, and it will be almost impossible to find the culprit. There are various applications to encrypt and sync the private keys across various devices to recover the private key (Menčik, 2016).

2.6.5.4 System hacking.

Blockchain records or data cannot be easily modified or altered. However, the code-base and system which implements blockchain can be modified because, depending on the company or organization, the blockchain codebase can be based on open source. For example, the most popular blockchain applications Bitcoin and

Ethereum are open source. Therefore, any user can contribute to the development of these applications and if these contributors provide vulnerable code or there is human-error in the code-base because of the contributor, it might possibly end up in the production system which in turn might cause system hacking. Beside this, a company can fork the blockchain codebase for their own use. If this codebase is poorly maintained, or outdated, it might end up getting hacked (Underwood, 2018).

2.6.5.5 Illegal activities 3 Blockchain.

The pseudo anonymity, immutable transaction, and decentralized property of the blockchain makes it difficult to monitor and track transactions on blockchain. Moreover, the technology itself is in the early stages of production and the essential regulations for using blockchain application are on early stages. Hence, the system can be misused for money laundering, illegal movement of funds. For example, Silk Road, a website to buy and sell illegal drugs used bitcoin for its payments (Vishnia, 2020).

2.7 Blockchain applications.

Blockchain can be used in different application domains such as financial, non-financial, insurance, Internet of Things (IOT), healthcare, Internet, cryptocurrency. Some of the financial applications are Medici, Blockstream, Bitshares and non-financial applications are Stampery, Ascribe, Block Notary. Everldger is Insurance application and IOT Filament, ADEPT platform is IOT applications. Examples of Internet applications are Namecoin, Ethereum. Examples of Cryptocurrency applications are Namecoin, bitcoin (Smits, 2020). Some of the most popular applications such as bitcoin, Namecoin and Ethereum are described below.

2.7.1 Bitcoin.

Bitcoin is the most popular application developed on blockchain. It wasnfirst proposed by Satoshi Nakamoto in 2008 in the paper 'Bitcoin: A Peer-to-Peer Electronic Cash System' that describes a P2P method of sending electronic cash from one person to another without involving any trusted third party. The electronic cash is termed as cryptocurrency. It relies on public-key cryptography to identify users, hash cash algorithm for proof of-work to detect double spending and consensus algorithm to reach the common agreement on blocks getting added to blockchain. Bitcoin was developed to address the current financial challenges. The current finance system is strongly tied with trusted third parties such as banks, credit card companies who relay and process the financial transactions. The trusted third parties validate, safeguard transactions, and persist the transaction history which later can be used as proof of financial transactions to avoid frauds. However, the economic collapse in 2008 has shown that these trusted parties can create economic bubble resulting to disastrous economic consequences (Garratt, 2018).

2.7.2 Namecoin

Namecoin is an alternative cryptocurrency (altcoin) based on BitDNS protocol with intend to enable censor-resistance domain name system outside the control of any single entity. The system uses blockchain to manage domain name lookup instead of central authority like Internet Corporation for Assigned Names and Numbers (ICANN). The latter requires of much trust on a central authority and represent a single point of failure. Namecoin was the first fork of the bitcoin codebase with its own blockchain. It uses Bitcoin core features such as proof-of-work, block creation time and transactions operations with additional features of a name/value store. The name/value store is a blockchain transaction database where user can store arbitrary identity data such as username, email address or website identity. It has been primarily used for the website identities and it enables registration and domain-name resolution for top-level domain (TLD). bit.

Namecoin squares Zooko Traingle, meaning it makes it possible to have domain name which is human-readable, decentralized and authenticated. Human-readable means a user can pick a name. Decentralized means there is no central trusted party or single point of failure. Authenticated means a strong sense of ownership using cryptographic keypair. Until 2011, designing a system which would exhibit all these three properties was impossible. However, Namecoin was the first system to provide naming system that offered all three properties. Namecoin cryptocurrency is called name-coin and its unit is represented by NMC. Furthermore, a user can register for. bit TLD at dotbit.me with very a small fee of 0.01 BTC or 5 or 20 NMC at the time of writing. Moreover, like bitcoin, Namecoin is also limited to 21 million coins. The first Namecoin block was mined in April 2011 and as of the time of writing over 327,299 blocks (which equals to 13,347,132 NMC) have been mined (Kang, 2017).

Based on Namecoin, various applications such as One Name and Block stack has been developed. One Name utilizes the Namecoin blockchain to record data about its members while Block stack provides similar features with Bitcoin blockchain and introduces separation of control and data plane and additional support of deploying decentralized server-less applications (KIRILLOVA, 2018).

2.7.3 Ethereum

Ethereum is the second most popular blockchain application. It was developed to address the weaknesses of the bitcoin. The weaknesses are bitcoin script that has the limit of small instructions and is non-Turing complete. The script is more centered toward bitcoin use case. Developing applications using Bitcoin script requires developers to fork the bitcoin core code-base and add the logic for their own use cases. The forking is time consuming and difficult to maintain. Thus, to address these challenges, Ethereum was developed. The Ethereum provides a platform for programmers to build applications on top of the blockchain called an Ethereum blockchain. It was first proposed in late 2013 by a Bitcoin programmer named Vitalik Buterin in the Whitepaper ’Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform’. This thesis proposes Turing-complete programming language for writing scripts (smart contracts) and Ethereum Virtual Machine (EVM) to execute the smart contracts and transactions. An Ethereum user can create smart contracts and upload them to the Ethereum Blockchain with a small fee. Other Ethereum users can access these contracts by remote procedure calls provided by Ethereum Application Program Interface (API). The contracts can store data, send transactions, and interact with other contracts. The contracts are executed in bytecode. Once contracts are uploaded to the blockchain, they are stored, executed, and interpreted by EVM. EVM requires a small amount of fees to execute transactions. These fees are called gas and the amount of gas depends on the size of instruction. The longer the contract instructions, the more gas is required. In addition, Ethereum has own cryptocurrency called ether and it is presented by the abbreviation ETH. Ether is a type of token that powers applications on the decentralized Ethereum network. The smallest unit of ether is Wei. One ether is equal to 10^{18} Wei. Users can use the Ethereum exchange to change the physical or normal money to ether (Leonhard, 2019).

3.0 CHAPTER 3: RATIONAL BEHIND CHOICE OF TECHNOLOGY.

As discussed, has various vulnerabilities and limitations. These vulnerabilities have caused user data hijack and breaches, identity theft and financial loss. These issues are becoming more common and frequent. This has sparked the security concerns over the current atheization framework. The end-users are becoming more concerned about their digital identity and privacy. Beside these issues, repeated user registration across different services is inconvenient. Multiple registrations increase the vulnerabilities of the user data. Thus, an alternative solution is required to address these challenges.

As discussed, Blockchain is technology based on the P2P, consensus protocol and digital signatures. The P2P network is the blockchain network, which is by design decentralized, distributed with no single point of failure. The consensus protocol ensures that a transaction (user A sends $1 to user B) happens only once. This transaction is added to public distributed ledgers which cannot be reverted. Also, anyone in the network can validate and verify this transaction. This makes the system transparent and reliable (Tarkhanov, 2020).

The user issues their identity with public-private key cryptography. This identity uses digital hash algorithm which is almost impossible to be cracked by current technologies. Moreover, the user's identity as well as data signed by the user can be verified and validated by anyone in the network. But the transactions signed by the private key can be only viewed by the owner. Thus, blockchain ensures the user identity is uncrackable and the user has complete ownership of user data as well anonymity over the network. Therefore, blockchain decreases the chances of hacking as the data is not shared with the central server. The user data is kept by the user and that data is protected by the latest hash algorithm. This hash algorithm is most advanced hash algorithm and has not been cracked yet. It is easy to use blockchain technology across multiple services. Moreover, user data is only with the user and not sent to central server. These features give the data ownership to the user than service providers. Blockchain also decreases the chances of the user data breach. The breach is only possible with user consent or carelessness. The providers cannot share data with third party organizations as they do not have user data and no control over their data. This technology has its own vulnerability as described. Also, it is not scalable compared to the current system because of the transactions time. But, despite these drawbacks, it is more secure, easy-to-use, trustworthy, reliable, fault-tolerant than the current AAA system as described. It is difficult to develop the applications for blockchain (e.g. bitcoin blockchain) because of the technical depth and architecture. The blockchain architecture is different than most of the existing AAA systems. The technology is in the early stages of the development and lacks the proper development instructions and support. A developer needs to clone the whole blockchain repository and develop the application on top of it with blockchain scripts which is cumbersome and difficult to deploy as described. The user also must maintain the blockchain and ensure it has the latest changes. The blockchain application must be modified according to these new latest blockchain changes. The blockchain was primarily designed for the Bitcoin. Thus, building other types of application based on this design is difficult and challenging. Ethereum Blockchain is the blockchain platform. It has been designed to develop the blockchain applications on top of the Ethereum blockchain using smart contracts. The smart contracts are written using high level solidity language. The language is easy to learn and write. It is a like JavaScript scripting language. The contracts can be easily deployed to private, test or main network. It is interpreted by the Ethereum Virtual Machine (Kondyrev, 2017).

Therefore, on one hand, Ethereum platform hides most of the technical depth of the blockchain and allows the developer to concentrate on writing his application logic while, on the other hand, it makes the application deployment painless. The large Ethereum community provides active support for the possible issues. For these reasons, Ethereum Blockchain was selected for the prototype development and testing.

4.0 CHAPTER 4: PROTOTYPE DESIGN& PROTOTYPE IMPLEMENTATION.

The design of prototype is described in this section. The reasoning behind the design and development of prototype as well as the software architecture is explained. It is worth noting that, this prototype is a very basic proof of concept with no strictly defined expectations and the result may vary depending on the which blockchain network is selected.

4.1 Need for Prototype

The goal of this thesis is to develop a proof-of concept on how services providers use one common identity backend to authenticate and authorize users. The proof-of-concept should also provide ownership of user data to the users rather than to services providers. The users should be able to login, register and pay their invoices without sharing their private data. The actual need of prototype comes from services providers having their own identity backends. These identity backends are a single point of failure. They are also vulnerable to different types of attacks and back-door of user data leaks as described.

Thus, services providers need one common distributed decentralized backend. This backend can authenticate and authorize services users with no single point of failure and decrease the possibility of attacks and user data leakages via back-doors. Another essential purpose of the prototype is the services user's data ownership. The user data is currently owned by services providers once services users register for services. Thus, service users need a system for their data ownership. The users should be able to use multiple service providers with same identity without sharing their private data. This ensures users are able pay their invoices without sharing their user data (Patil, 2018).

4.2 Software Architecture

The software architecture presented in the below Figure shows the complete solution on how the services providers could leverage Ethereum blockchain technology for a common identity backend. The solution also presents how users could use blockchain technology for data ownership and pay their invoices without sharing their private financial data. Developing the complete solution as shown in the below Figure is out of scope for this thesis. However, the whole complete solution has been discussed from the architecture perspective. Most crucial parts: implementing smart contracts to the blockchain network was implemented and relation between providers, blockchain, smart contract and users was described in detail. The software architecture has four types of participants: infrastructure providers, service providers, Ethereum Blockchain network and a user as shown in the Figure. Infrastructure providers are providers which provide infrastructure service es such as computing, storage, network to the service providers.

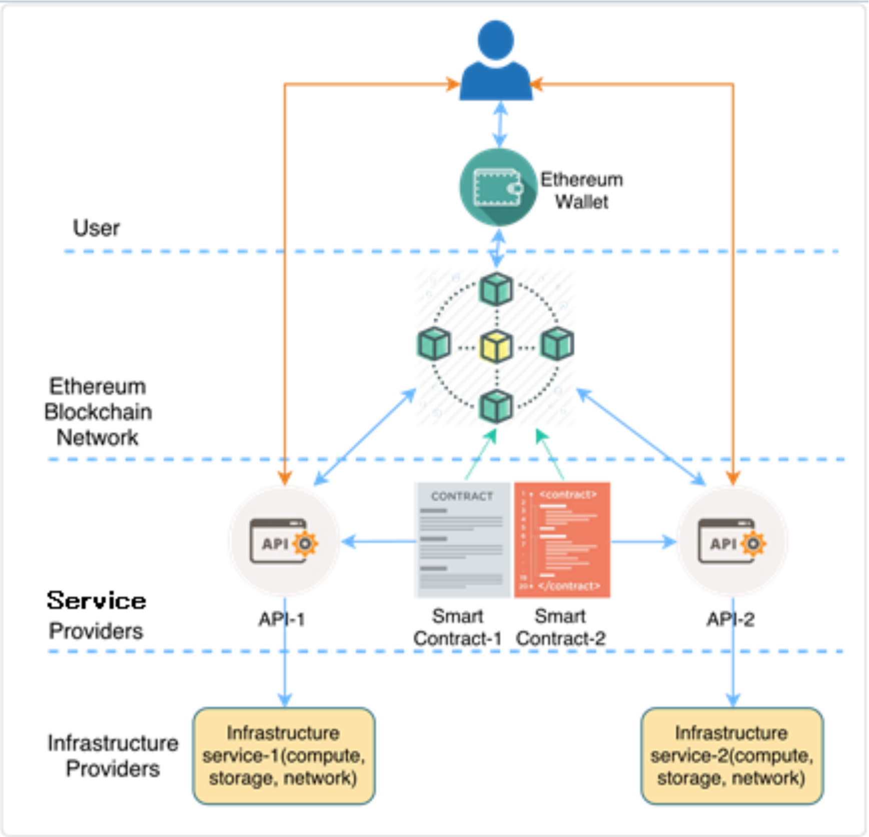


Figure 9. Blockchain Software Architecture

For simplicity, a user, two infrastructure services 1 and 2 as well as two service providers 1 and 2 with their respective smart contracts are considered as shown in the Figure above. Additionally, the figure can be easily extended for multiple users by adding more users with their own Ethereum wallet. Infrastructure services interact with their respective service providers. Each service providers define their authentication and authorization logic with smart contracts. These contracts are deployed to the Blockchain network via their respective API. The user creates their identity with the Ethereum wallet that generates set of private and public key. The wallet stores the private key while deploys the public key to the Ethereum network. Now, the users can access the infrastructure service through the services provider API which authenticates and authorize the user with the Blockchain network. Multiple service providers connect to common identity backend. The users also connect to the same identity backend. For the user to use service, she/he can leverage the blockchain for identity without providing any private information to the service providers. Also, a user does not need to register for new services provider to use their services. According to this architecture any service provider can basically integrate or connect to the existing infrastructure provider and offer their services without requiring users to register with their services.

4.3 Flow diagram

The flow diagram describes how the user interacts with a service provider smart contract which is deployed to the blockchain network. For simplicity, we assume that services resources are used. As discussed above, services resources are out of scope of this thesis. The key components in the figure are the smart contracts deployed by the services provider and a user able to make transactions using the smart contract in the blockchain network.

The user must be authenticated and authorized to access the services resources as well as to pay the invoice in cryptocurrency. There are two types of authentication. The user and service provider both are authenticated against the blockchain to use services resources and execute the transactions, respectively. The user authenticity is proved recovering the user public key from the message signature which is signed with the user private key. The services provider authenticity is proved as all the transactions executed by the services provider are signed with the services provider private key. These signatures are by default verified by the blockchain before executing the transactions. Hence, this ensures, on one hand, the user is legitimate user and only the right service provider can run the transaction. The services provider authorizes the user by checking if the user address is valid and exists on the blockchain. If the answer is positive, the services provider adds the user to its blockchain address database and marks the user as authorized to access its resources. Furthermore, only public key of the services provider as well as the public key of the services user is distributed to the blockchain. Hence, the prototype maintains the anonymity of the user and provider. Additionally, the user can pay the invoice with the cryptocurrency without providing their bank details.

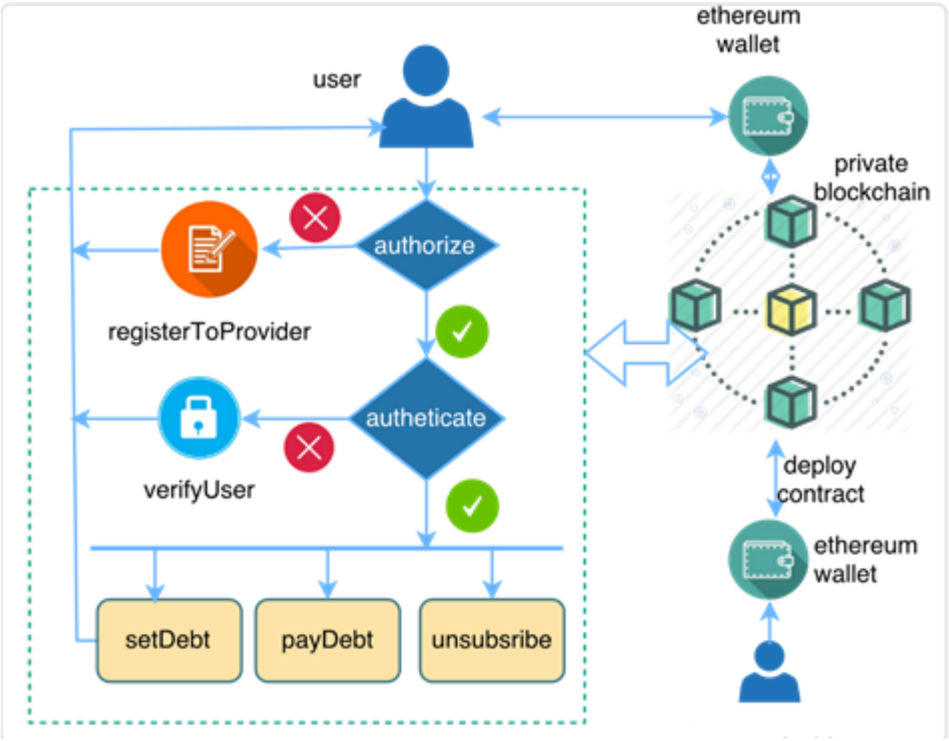


Figure 10. System overall Flow diagram.

First the services provider deploys the contracts to the private blockchain network with ethereum wallet. Meanwhile, a services user creates his/her digital identity using ethereum wallet. Now, the user can access services. If the user is authenticated and authorized, she/he can access the services. Otherwise, the user needs to do the action registerToProvider. On successful authorization, the user can access services. The user identity still needs to be verified. On success, the prototype proves the authenticity of the user and the user is authorized and authenticated to access the resources. The services providers keep the track of its resource usage by the user. Eventually the provider sets the debt to the user using setDebt method. The user can pay this debt in ether with payDebt method. Furthermore, the user can also de-register from the services provider. This deactivates user from services provider and the user needs to register again to access the services.

4.4 Prototype implementation.

This section describes the hardware and software components required for implementing the prototype. It also describes the smart contracts, prototype environment setup and how the prototype was executed. The main idea is, after reading this section, it would be possible to setup the development environment and execute the prototype.

4.4.1 Hardware Components

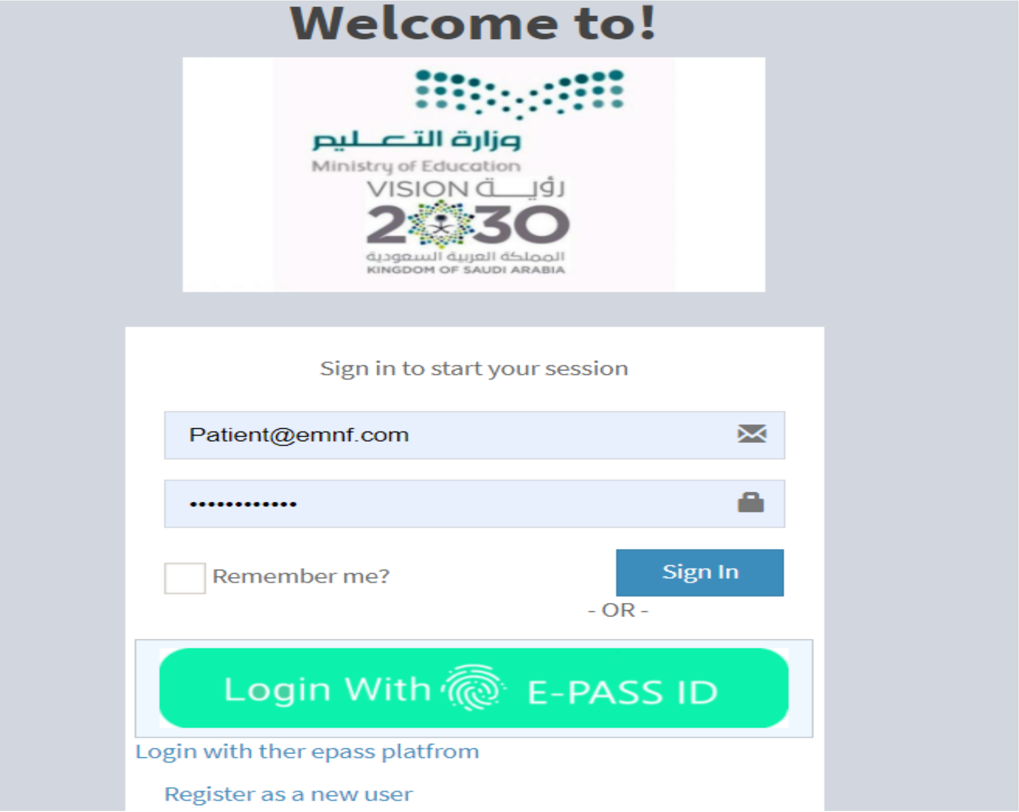
The prototype was developed and tested on MacBook Pro, Mid 2010 computer. The computer has 2,4 GHz Intel Core 2 Duo processor, 8 GB memory and operating system macOS Sierra version 10.12.3.

4.4.2 Software Components

This section describes the software components used for prototype development and testing. The components are entity platform, API gateway and E-pass platform.

4.4.3 Entity platform.

The selection fell on the platform of the Saudi Ministry of Education.



Patient@emnf.com

•••••••••••••

Figure 11. Saudi Ministry of Education platform login page..

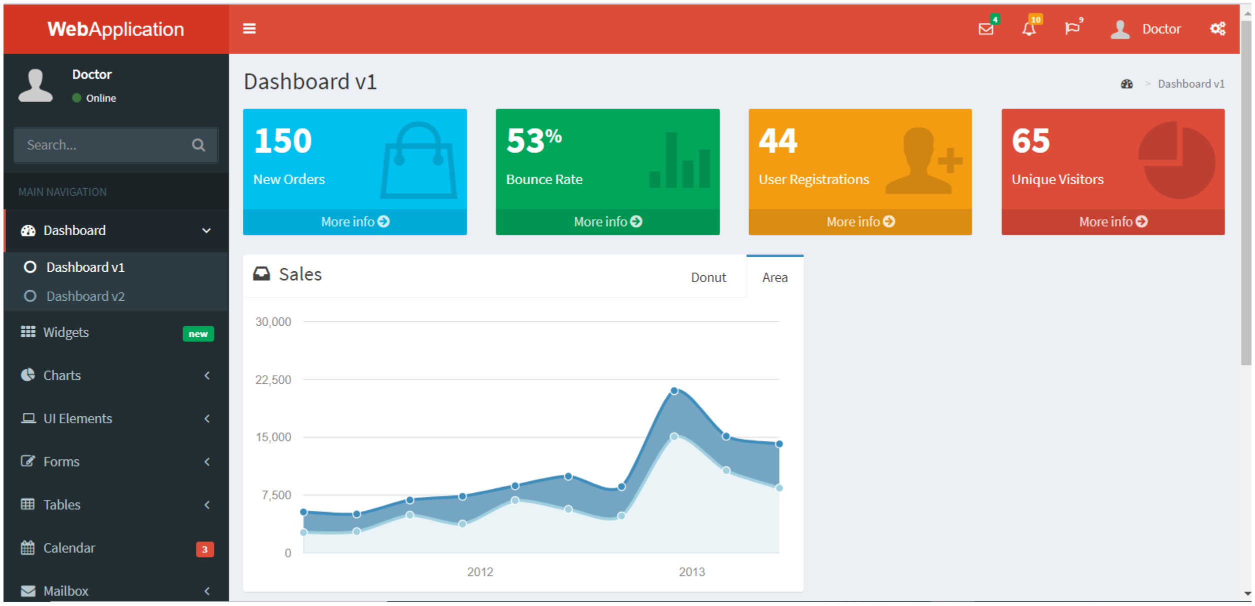
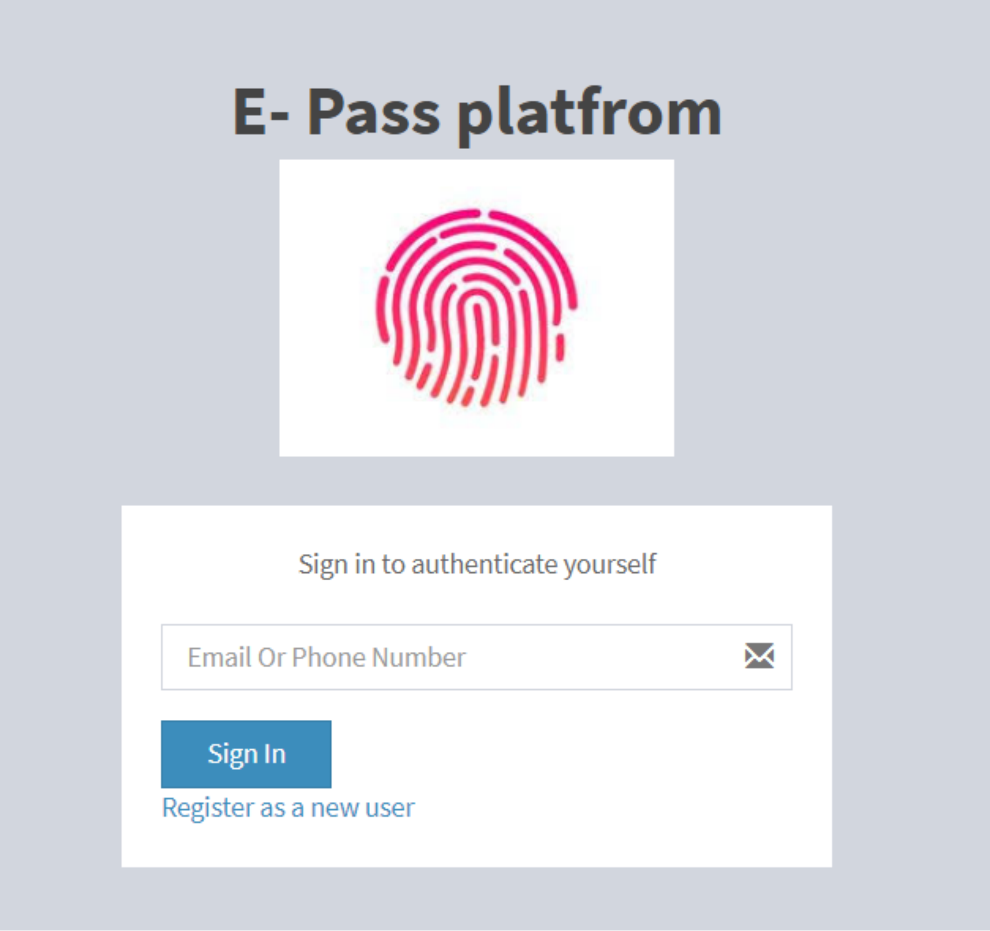


Figure 12. Saudi Ministry of Education dashboard.

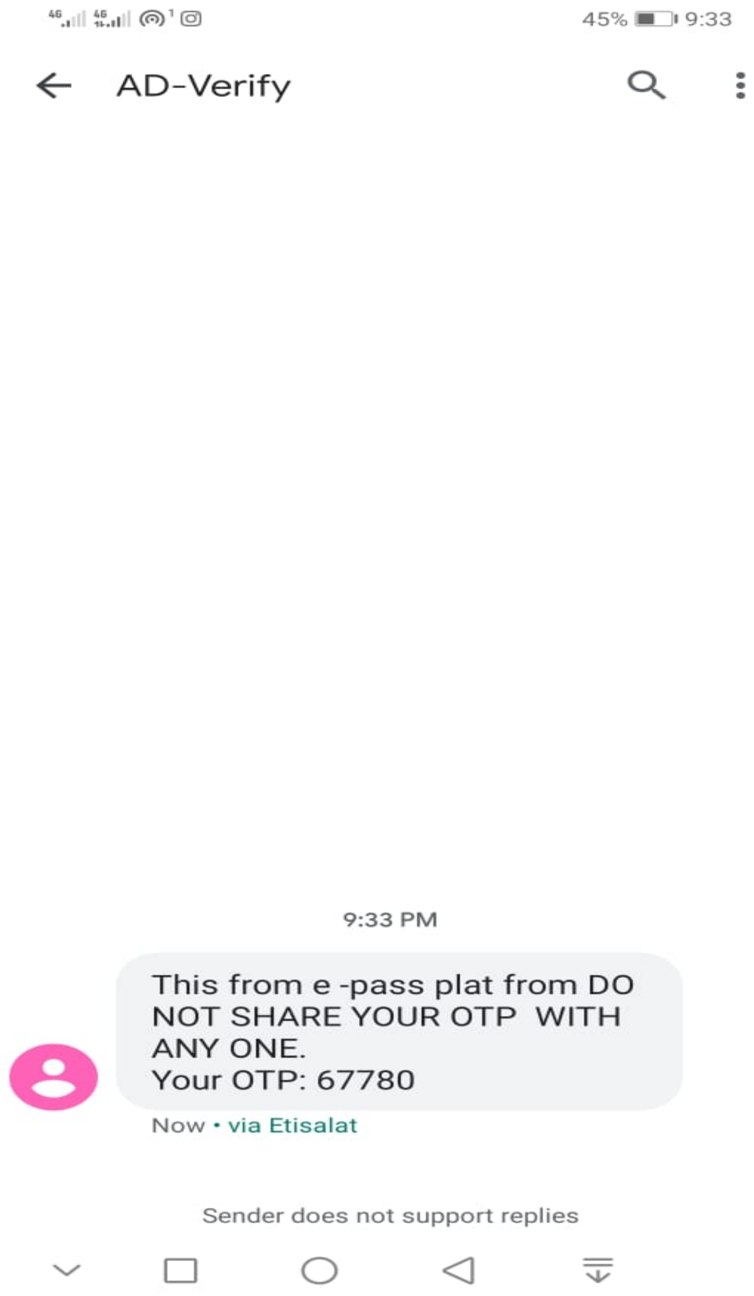
4.4.4 E-pass platform.



E- Pass platfrom

Sign in to authenticate yourself

Figure 13. E-pass platform authentication.



9:33 PM

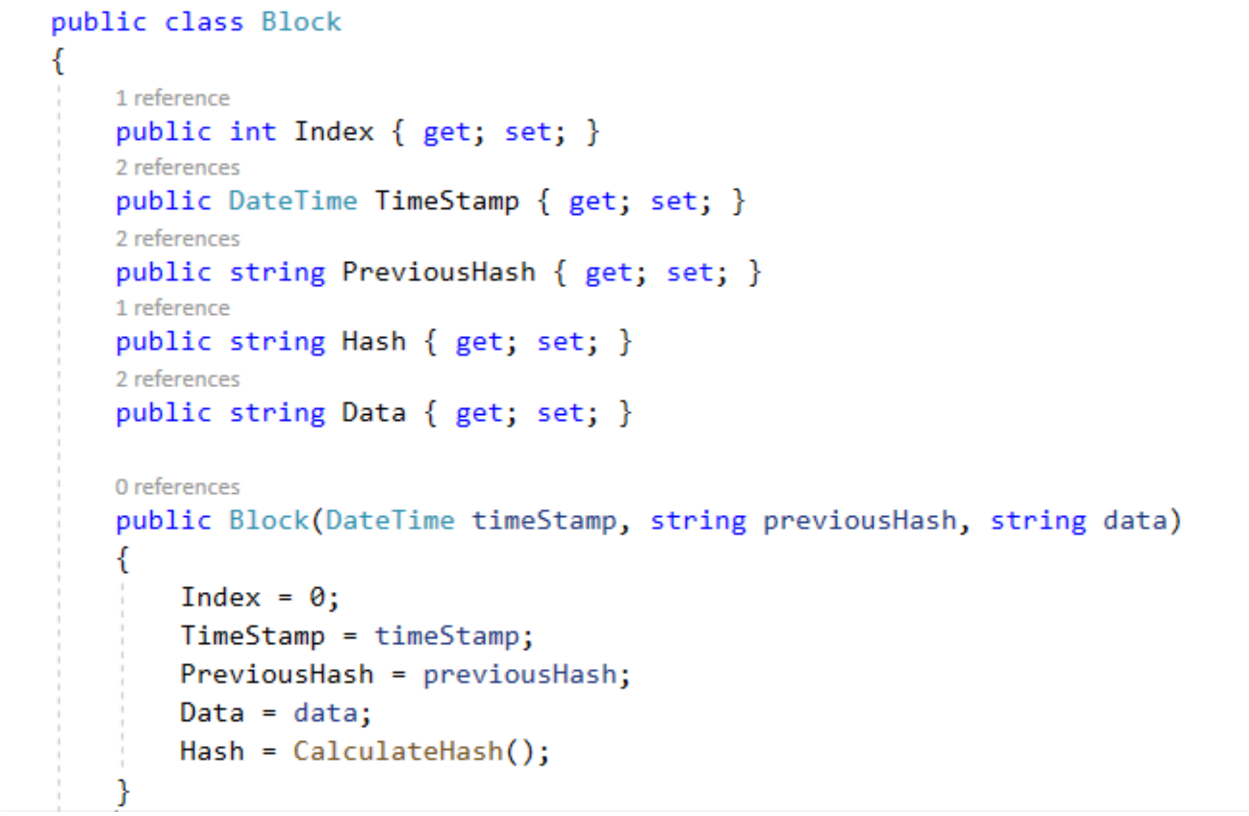
This from e -pass plat from DO NOT SHARE YOUR OTP WITH ANY ONE.
Your OTP: 67780

Now • via Etisalat

Sender does not support replies

Figure 14. E-pass OTP.

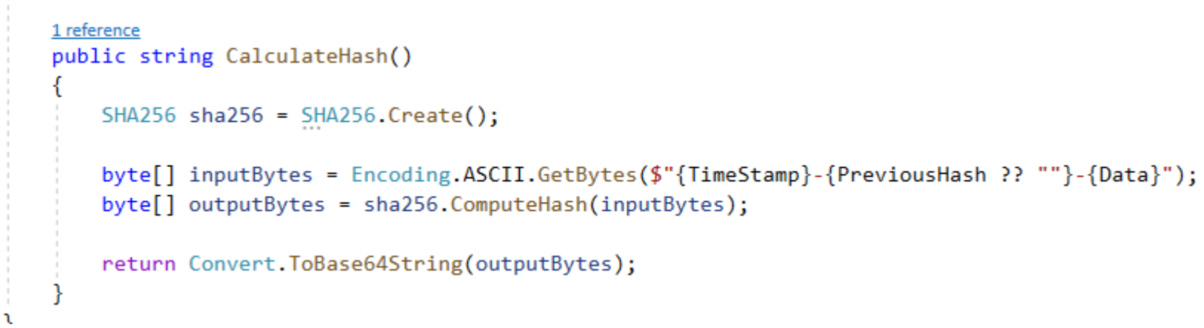
4.4.5 Blockchain codes.



```
public class Block
{
    1 reference
    public int Index { get; set; }
    2 references
    public DateTime TimeStamp { get; set; }
    2 references
    public string PreviousHash { get; set; }
    1 reference
    public string Hash { get; set; }
    2 references
    public string Data { get; set; }

    0 references
    public Block(DateTime timeStamp, string previousHash, string data)
    {
        Index = 0;
        TimeStamp = timeStamp;
        PreviousHash = previousHash;
        Data = data;
        Hash = CalculateHash();
    }
```

Figure 15. Block Model.

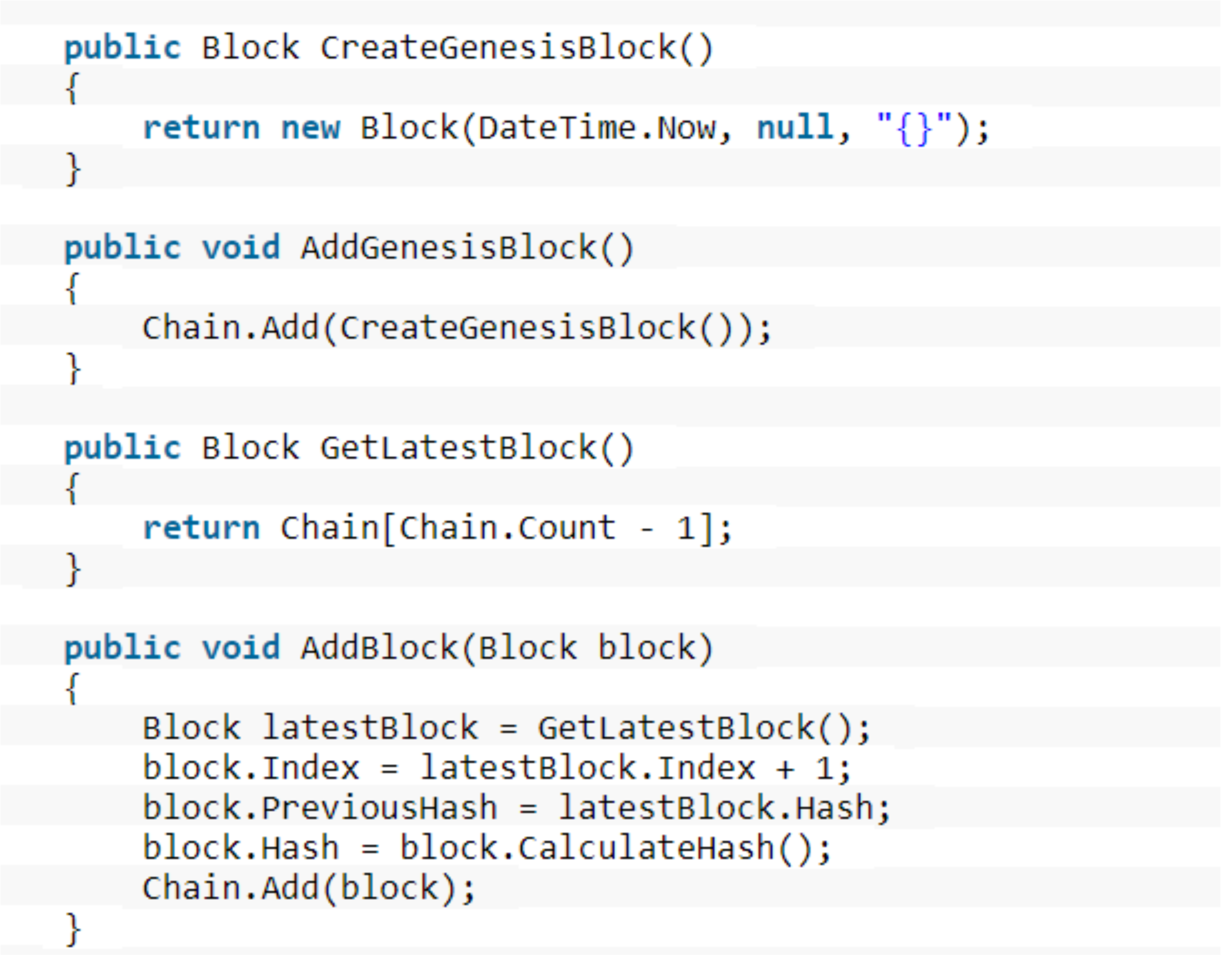


```
    1 reference
    public string CalculateHash()
    {
        SHA256 sha256 = SHA256.Create();

        byte[] inputBytes = Encoding.ASCII.GetBytes($"{TimeStamp}-{PreviousHash ?? ""}-{Data}");
        byte[] outputBytes = sha256.ComputeHash(inputBytes);

        return Convert.ToBase64String(outputBytes);
    }
```

Figure 16. Calculate Hash method.



```
public Block CreateGenesisBlock()
{
    return new Block(DateTime.Now, null, "{}");
}

public void AddGenesisBlock()
{
    Chain.Add(CreateGenesisBlock());
}

public Block GetLatestBlock()
{
    return Chain[Chain.Count - 1];
}

public void AddBlock(Block block)
{
    Block latestBlock = GetLatestBlock();
    block.Index = latestBlock.Index + 1;
    block.PreviousHash = latestBlock.Hash;
    block.Hash = block.CalculateHash();
    Chain.Add(block);
}
```

Figure 17. Helper methods.

5.0 CHAPTER 5: PRIMARY RESEARCH RESULTS.

We have conducted a survey that includes more than 40 people which are interested in the technology sector and have Internet skills as will. Most of these people have financial and social interests that depend on the smart services of the public and the private sectors alike. We found that more than 20 people use online services at least 12 times a month, which is equivalent to 3 transactions per week, and 10 of them are more clearly using the services 5 times per month, which is a relatively high rate, hence, we conclude that the smart services sector that be provided through internet channels is a vital and important sector, hence the security aspect should be strengthened in order to be comfortable for the stockholders. When asked whether they were hacked before or their log-in information has spread, 5% of them have been hacked, meaning out of every 40 people there are 2 who have been hacked, which is a worrying number for the pioneers of smart transactions over the Internet, even for Internet users in general, and 14 of them been a victims where their confidential information has been rife and their digital identity have been exposed, this forcing them to not trust the reliable and unreliable sites on which they conduct their daily transactions, this would confuse this vital sector and reduce the possibility of the development of this sector and facilitate the transactions. And the good news that we received from the volunteers who responded to the questionnaire, they are satisfied with the unified digital identity that will protect their identity on the Internet, it won the approval of most of the public, and more than ninety percent, strongly agreed on the idea after it was discovered to them that the technology used in this feature is the blockchain technology. Seventy percent of them who acquire technical skills are encouraged to launch this platform as soon as possible to benefit from its services, overcome authentication difficulties and stop the severe digital impersonation operations either. The platform also received the overall improvement of its speed in login or signup with the service provider plat from, the results of the survey were generally satisfactory, and as we see below the results of two questions.

6.0 Conclusion and Future work

6.1 Conclusion.

This thesis has proposed an AAA solution based on Ethereum blockchain and made contribution in AAA for the online services environment. The problem has been solved and goal has been achieved. The architecture of proposed solution consists of three main components: Ethereum wallet, smart contracts and Ethereum blockchain. Ethereum wallet is responsible for authentication of the user against Ethereum blockchain by creating private-public key. The public key is distributed across the blockchain network and private key is kept secret with the user and it is pass-phrase protected. This key-pair acts as user identity where the network can verify and validate user authenticity by user’s public key. The smart contracts have the core logic of user authorization and the assisting logic of authentication as well as accounting and de-registration of user. The contracts in-charge are the cloud providers who develop and deploy it to the blockchain.

6.2 Future work.

As a prototype, AAA with Ethereum Blockchain is merely rock solid. Since the objective of this thesis was just to develop and test basic proof-of-concept, there are a lot of features and improvements to be done. The remaining components from the architecture diagram API could be developed and a real blockchain be added to complete the solution. One of the future works is to develop a smartphone application and integrate it with the solution to provide a higher level of one-time password. The user can open the application and receive a notification for authentication and upon authentication, the user can enter the service provider’s system and complete the promoted services and dispense with the OTP.

References:

1. Akanbi, A. B. (2020). A STACKED ENSEMBLE FRAMEWORK FOR DETECTING MALICIOUS INSIDERS. *International Journal of Innovative Research in Computer Science & Technology*.
2. Banach, R. (2020). Blockchain applications beyond the cryptocurrency casino: The Punishment not Reward blockchain architecture. *Concurrency and Computation: Practice and Experience*.
3. Charles, W. (2019). Blockchain Compliance by Design: Regulatory Considerations for Blockchain in Clinical Research. *Frontiers in Blockchain*, 27-77.
4. CHEN, Y.-Y. (2015). Pointer Logic for Verification of Pointer Programs. *Journal of Software*.
5. Cheng, L. (2017). Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*.
6. Dean, D. H. (2014). The benefit of a trustworthy face to a financial services provider. *Journal of Services Marketing*, 771-783.
7. Garratt, R. (2018). BITCOIN 1, BITCOIN 2, ...: AN EXPERIMENT IN PRIVATELY ISSUED OUTSIDE MONIES. *conomic Inquiry*.
8. Gauravaram, P. (2018). The legal and practical implications of recent attacks on 128-bit cryptographic hash function. *First Monday*.
9. Goodell, G. (2018). A Decentralized Digital Identity Architecture. *Frontiers in Blockchain*.
10. Ha, N. (2017). Efficient Defense Mechanism against Sybil Attack in Wireless Sensor Network. *International Journal Of Engineering And Computer Science*.
11. Hawkins, J. (2014). Measuring Potential Vulnerabilities in Emerging Market Economies. *SSRN Electronic Journal*.
12. Hoffmann, T. (2019). Blockchain, Smart Contracts und Recht. *Informatik Spektrum*.
13. Jal, A. (2018). ecure The Data In Multi Cloud Using Erasure Code And Merkle Hash Tree Algorithm. *International Journal of Recent Trends in Engineering and Research*.
14. Kang, H. (2017). Safe and convenient personal authentication method using Moiré 3D authentication based on biometric authentication. *Cluster Computing*, 2017-2026.
15. KIRILLOVA, E. A. (2018). Bitcoin, Lifecoin, Namecoin: The Legal Nature of Virtual Currency. *Journal of Advanced Research in Law and Economics*.

16. Kondyrev, D. O. (2017). Ethereum-Based Tender System. *Vestnik NSU. Series: Information Technologies*.
17. Korshid, A. (2016). Revolution under attack: the Forqan Group of Iran. *Choice Reviews Online*, 53-1940-53-1940.
18. Leonhard, R. (2019). Decentralized Finance on the Ethereum Blockchain. *SSRN Electronic Journal*.
19. M Smits, J. H. (2020). Blockchain Applications and Institutional Trust. *Frontiers in Blockchain*.
20. Madhusudhana. (2019). Authentication for Computer Communication > An Overview of Authentication for Computer Communications. *Advanced Computing and Communications*, 13-55.
21. McConnell, P. (2016). Blockchain Examining the Technical Architecture. *ITNOW*.
22. Menčik, D. (2016). Identity Theft: A Thought Experiment on the Fragility of Identity. *Conatus*.
23. Mets, T. (2017). Time of signing in the Estonian digital signature scheme. *Digital Evidence and Electronic Signature Law Review*.
24. Nia, J. (2017). Attack on government: fear, distrust, and hatred in public life. *Choice Reviews Online*, 42-5551-42-5551.
25. Patil, P. (2018). Authentication through Claims-Based Authentication. *International Journal of Trend in Scientific Research and Development*, 2664-2666.
26. Pech, S. (2017). Copyright Unchained: How Blockchain Technology Can Change the Administration and Distribution of Copyright Protected Works. *SSRN Electronic Journal*.
27. Polese, F. (2019). Smart City as a Service System: A Framework to Improve Smart Service Management. *Journal of Service Science and Management*, 1-16.
28. Riva, G. M. (2020). What Happens in Blockchain Stays in Blockchain. A Legal Solution to Conflicts Between Digital Ledgers and Privacy Rights. *Frontiers in Blockchain*, 44-67.
29. Sandhu, R. (2019). Password Less Authentication. *Indian Journal of Science and Technology*, 43.
30. Sandner, P. (2019). Speciality Grand Challenges: Blockchain. *Frontiers in Blockchain*, 56-86.

31. Shackelford, S. (2016). Block-by-Block: Leveraging the Power of Blockchain Technology to Build Trust and Promote Cyber Peace. *SSRN Electronic Journal*.

32. Smits, M. (2020). Blockchain Applications and Institutional Trust. *Frontiers in Blockchain*.

33. Tarkhanov, I. (2020). A method of data synchronization with Ethereum blockchain. *Artificial societies*.

34. Trappe, W. (2018). Cybersecurity: A New Open Access Journal. *Cybersecurity*, 1.

35. Treiblmaier, H. (2019). Toward More Rigorous Blockchain Research: Recommendations for Writing Blockchain Case Studie. *Frontiers in Blockchain*, 45-56.

36. Underwood, E. (2018). Hacking the immune system. *Knowable Magazine*.

37. Vishnia, G. R. (2020). AuditChain: A Trading Audit Platform Over Blockchain. *Frontiers in Blockchain*, 34-76.

38. Zwitter, A. (2020). Decentralized Network Governance: Blockchain Technology and the Future of Regulation. *Frontiers in Blockchain*, 43-98.

39. Houbing Song, Glenn A. Fink, and Sabina Jeschke, Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications. ISBN: 978-1-119-22604-8, Chichester, UK: Wiley-IEEE Press, 2017, pp. 1-472.

40. I. Butun, P. Österberg and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks and Countermeasures," in IEEE Communications Surveys & Tutorials. doi: 10.1109/COMST.2019.2953364

41. W. Li, et al., "ART: An Attack-Resistant Trust Management Scheme for Securing Vehicular Ad Hoc Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 960-969, April 2016.

42. I. Butun, M. Erol-Kantarci, B. Kantarci and H. Song, "Cloud-centric multi-level authentication as a service for secure public safety device networks," in IEEE Communications Magazine, vol. 54, no. 4, pp. 47-53, April 2016, doi: 10.1109/MCOM.2016.7452265.

43. Yifan Tian, Jiawei Yuan, Houbing Song, Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones, Journal of Information Security and Applications, Volume 48, 2019, 102354, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2019.06.010.

44. Jian Chen, Zhihan Lv, Houbing Song, Design of personnel big data management system based on blockchain, Future Generation Computer Systems, Volume 101, 2019, Pages 1122-1129, ISSN 0167-739X, https://doi.org/10.1016/j.future.2019.07.037.

45. Y. Tian, J. Yuan and H. Song, "Secure and Reliable Decentralized Truth Discovery Using Blockchain," 2019 IEEE Conference on Communications and Network Security (CNS), Washington DC, DC, USA, 2019, pp. 1-8, doi: 10.1109/CNS.2019.8802712.

46. Y. Liu, J. Wang, H. Song, J. Li and J. Yuan, "Blockchain-based Secure Routing Strategy for Airborne Mesh Networks," 2019 IEEE International Conference on Industrial Internet (ICII), Orlando, FL, USA, 2019, pp. 56-61, doi: 10.1109/ICII.2019.00021.

47. M. Hanif and H. Song, "Blocks' Network: Redesign Architecture Based on Blockchain Technology," 2019 IEEE International Conference on Industrial Internet (ICII), Orlando, FL, USA, 2019, pp. 34-39, doi: 10.1109/ICII.2019.00017.

48. M. Albalawi and H. Song, "Data Security and Privacy Issues in Swarms of Drones," *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Herndon, VA, USA, 2019, pp. 1-11, doi: 10.1109/ICNSURV.2019.8735133.

49. L. A. Tawalbeh, W. Bakheder and H. Song, "A Mobile Cloud Computing Model Using the Cloudlet Scheme for Big Data Applications," *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, Washington, DC, 2016, pp. 73-77, doi: 10.1109/CHASE.2016.40.

50. S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song and H. Yu, "SAMADroid: A Novel 3-Level Hybrid Malware Detection Model for Android Operating System," in *IEEE Access*, vol. 6, pp. 4321-4339, 2018, doi: 10.1109/ACCESS.2018.2792941.

51. A. Ossamah, A. Meshari, A. Yazed and A. Norah, "Cloud Based Cyber Physical System for Factory Automation," *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, 2020, pp. 1-7, doi: 10.1109/WF-IoT48130.2020.9221312.